Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey

Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis and Robert Atkinson
Department of Electronic & Electrical Engineering
University of Strathclyde
Division of Computing and Mathematics
University of Abertay Dundee

E-mail: elike.hodo@strath.ac.uk
E-mail: x.Bellekens@abertay.ac.uk

   Abstract— Intrusion detection has attracted a considerable interest from researchers and industries. The community, after many years of research, still faces the problem of building reliable and efficient IDS that are capable of handling large quantities of data, with changing patterns in real time situations. The work presented in this manuscript classifies intrusion detection systems (IDS). Moreover, a taxonomy and survey of shallow and deep networks intrusion detection systems is presented based on previous and current works. This taxonomy and survey reviews machine learning techniques and their performance in detecting anomalies. Feature selection which influences the effectiveness of machine learning (ML) IDS is discussed to explain the role of feature selection in the classification and training phase of ML IDS. Finally, a discussion of the false and true positive alarm rates is presented to help researchers model reliable and efficient machine learning based intrusion detection systems.

   Keywords— Shallow network, Deep networks, Intrusion detection, False positive alarm rates and True positive alarm rates

1.0 INTRODUCTION

Computer networks have developed rapidly over the years contributing significantly to social and economic development. International trade, healthcare systems and military capabilities are examples of human activity that increasingly rely on networks. This has led to an increasing interest in the security of networks by industry and researchers. The importance of Intrusion Detection Systems (IDS) is critical as networks can become vulnerable to attacks from both internal and external intruders [1], [2].

An IDS is a detection system put in place to monitor computer networks. These have been in use since the 1980's [3]. By analysing patterns of captured data from a network, IDS help to detect threats [4]. These threats can be devastating, for example, Denial of service (DoS) denies or prevents legitimate users resource on a network by introducing unwanted traffic [5]. Malware is another example, where attackers use malicious software to disrupt systems [6].

Intrusion detection systems evolved as a response to these situations. Many IDS have been developed but have experienced the problem of false positive or false negative alarms. These are the false recognition of an attack by the IDS and increases the difficultly for network administrators to handle intrusion reports. Researchers in this field aim at developing IDS that have a high accuracy of detection and low false alarm rate [7]. Another problem of some existing IDS is their inability to detect unknown attack types. These IDS rely on the signatures of known attacks.

Human independent IDS that incorporate machine learning techniques have been developed as a solution to these problems. Machine learning IDS learns from normal traffic and abnormal traffic by training on a dataset to predict an attack by using classification. Several machine learning techniques have been successfully implemented as classifiers on IDS but present numerous flaws such as low throughput and high false detection rates [7].

The taxonomy (or classification) presented within this manuscript is highly relevant in the present study due to the highly diverse types of systems and attacks [8], [9]. The taxonomy will aid constructing two objectives: a clear description of the current state of IDS and the guide lines in which to explore the complexity of it [10], [11].

The paper aims to provide a clear description and guidelines needed to understand intrusion detection systems based on results from existing works in various papers. Its organisation gives a broader view of IDS and aims at addressing the concerns and solutions of shallow and deep learning IDS. The paper also compares previous works and their performance metrics.

The second objective of the paper is to present a survey and the classification of Intrusion Detection Systems, taxonomy of Machine Learning IDS and a survey on shallow and deep networks IDS.

2.0 INTRUSION DETECTION SYSTEMS

Intrusion detection systems are strategically placed on a network to detect threats and monitor packets. The IDS accomplishes this by collecting data from different systems and network sources and analysing the data for possible threats [12]. The functions of the IDS include offering information on threats, taking corrective steps when it detects threats and recording all important events within a network [13]. Figure 2 shows a model of Intrusion detection system
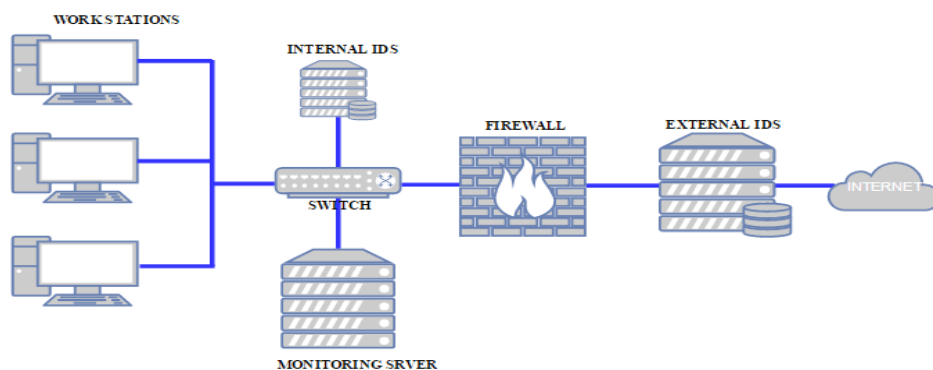

Figure 2 Intrusion detection system model

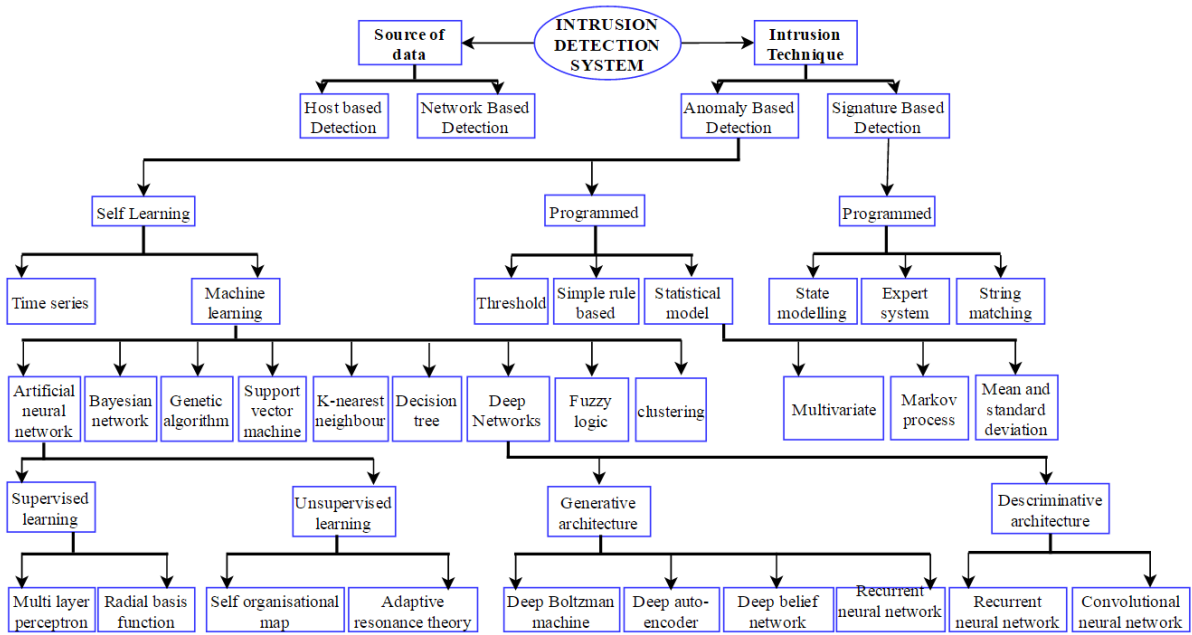## 2.1 Classification of Intrusion detection system

Figure 2.1 Classification of Intrusion Detection Systems

Different researchers have developed different classification representations. H.Debar *et al*[14], S.Axelsson *et al.[15]*, S.Amer *et al.*[16], C.Xenakis *et al.*[17], Hung-Ten Lia e*t al.*[18], have previously presented intrusion detection surveys and taxonomies. This research builds upon their work and introduces deep networks technique which all these referenced works do not describe. With the increasing value of big data, deep networks is an important element to capture in an IDS taxonomy The taxonomy presented within this work provides a fine-grained overview of the different machine learning techniques for intrusion detection systems as shown in Figure 2.1. The detection mainly depends on the source of data and intrusion technique used. The Source of data is the nodes that gather the information for analysis. The source can be either host or network based or both. These two sources are further discussed in section 2.1.1. The detection technique also differentiates between the anomaly based and signature based detection techniques. The different techniques that fall in the class of anomaly and signature based are important to develop a taxonomy that gives a broader view of techniques and the capabilities in detecting attacks. Section 2.1.2 and 2.1.3 explains anomaly based and signature based techniques in detail. These techniques operate either as self-learning or programmed. In Self-learning class, the system automatically learns whilst in programming class a user teaches the system. The classification presented Fig.2.1 will help draw conclusions on detection accuracy and false alarm rates of IDS techniques.

2.1.1 Host Based IDS (HIDS) Vs. Network Based IDS (NIDS)

Host based IDS were the first type of IDS to be implemented [19]. HIDS are software based products installed on a host computer that analyse and monitor all traffic activities on the system application files and operation system [20], [21]. The traffic activities gathered by the system application files and system [20] are called the audit trails [22]–[24]. HIDS has an

advantage of being able to detect threats from within by scanning through the traffic activities before sending and receiving data [25]. Its main disadvantage is that only monitors the host computer, meaning it has to be installed on each host [26].

Network based IDS are found at specific points on the network to capture and analyse the stream of packets going through network links, unlike the HIDS whose approach is to analyse each host separately [27] [28], [29]. It has the advantage of having a single system monitoring an entire network, saving time and cost of installing software on each host. NIDS main disadvantage is its vulnerability to any intrusion originating from the network targeting a system within the network.

Table I. Performance comparison of HIDS and NIDS [30]

Performance in terms of:	Host-Based IDS	Network-Based IDS
Intruder deterrence	Strong deterrence for inside intruders	Strong deterrence for outside intruders
Threat response time	Weak real time response but performs better for a long term attack	Strong response time against outside intruders
Assessing damage	Excellent in determining extent of damage	Very weak in determining extent of damage
Intruder prevention	Good at preventing inside intruders	Good at preventing outside intruders
Threat anticipation	Good at trending and detecting suspicious behavior patterns	Good at trending and detecting suspicious behavior patterns

2.1.2 Anomaly Based Detection

Anomaly Based Detection is a behavioural based intrusion detection system. It observes changes in normal activity within a system by building a profile of the system which is being monitored [31], [32]. The profile is generated over a period of time when the system is established to have behaved normally [33]. One advantage is that it offers the ability to detect attacks which are new to the system [34]. Anomaly detection is categorised into two, based on the way a normal profile of a system is specified.

1) Self-learning – The self-learning system operate by example with a baseline set for normal operation. This is achieved by building a model for the underlying processes with the observed system traffic built up over a period of time [15]. Self-learning systems are sub-divided into the following main categories: time series model and machine learning. Section 3 discusses machine learning technique in detail.

   - **Time series model** takes into account the sequence of observation in order of succession occurring in intervals of uniformity. If the probability of occurrence of a new observation at a time is negligible then it is considered a change in normal behaviour. Time series has an advantage of observing trends of behaver over a period of time and flagging it, if it notices a change in normal behaviour. It is an effective model when attacks are sequential over a period of time [35]. This model has a disadvantage of being more costly

computationally [36]. Auto regressive moving average (ARMA) is an example of time series model used as IDS.

In Predictive Modelling for Intrusions in Communication Systems using generalised autoregressive moving average (GARMA) and ARMA models by [37] *et al.* fitted ARMA(1,1) and GARMA(1,2;δ,1) time series models to 4 types of attacks (DoS, probe, U2R and L2R). The parameter estimation was done using Hannan-Rissanen algorithm, Whittle estimation and Maximum likelihood estimation and the point forecast obtained through Whittle estimation and maximum likelihood were close to the original value. The time series models were able to forecast the attack but the performance of GARMA (1, 2; δ, 1) in attack detection was better.

.

2) Programmed − A programmed model is when a system needs either a user or an external person to teach the system to detect changes in behaviour. The user decides the extent of abnormal behaviour in the system and flags an intrusion threat [35]. The programmed models are grouped into four categories: threshold, simple rule based, and statistical models.

- **Threshold models** can be considered as the simplest programmed descriptive statistical detector [15]. By correlating and analysing statistical data, a user can program the system at a pre-defined threshold of alarm on a statistical variable. Careful selection of the threshold is required to minimise the false alarm rate. Instances where the threshold set too high leads to a risk of missing an alarm when an intruder is performing a malicious action [38]. A typical example is setting an alarm if three attempts to login to a system are unsuccessful [39].

  In anomalous payload-based network intrusion detection [40], Ke Wang *et al*. presented a payload base anomaly detection by demonstrating the effectiveness on 1999 DARPA dataset and a live dataset collected form the US Columbia CS department network. In training phase, a profile byte frequency distribution was computed and the standard deviation of application payload connected to a single host and port. During the detection phase the Mahalanobis distance was used to calculate the similarity of new data against the pre-computed profile. The detector compared this measure against a threshold and generated an alert when the distance of the new input exceeded this threshold. It recorded nearly 100% in accuracy with 0.1% false positive rate for port 80 traffic.

- **Simple rule based** is an approach for detection is to monitor events within a system that trigger a rule either than what is considered as a normal behaviour for the user. One limitation of this model is its failure to detect a threat that is not programmed as a rule in the system [41]. RIPPER (repeated incremental pruning to produce error reduction) is an example of rule based model that builds a set of rules to detect normal behaviour and abnormal behaviour. R.Naidu *et al*[42] used KDDCup '99 dataset to compare the performance of RIPPER rule, decision tree(C5) and support vector machine(SVM). The

dataset was categorised into 3 (NORMAL, probe attacks and DoS attacks). The RIPPER rule algorithm went through two stages during the experiment. The first stage initialised the rule conditions and the second stage the rule optimisation. The algorithm obtained conditions in each rule to classify the testing data. The RIPPER rule recorded a total detection rate of 98.69%, C5 98.75% and SVM 98.63%.

- **Statistics model** collects data in a profile**.** Analysing the profile of normal statistical behaviour gives a description shown by the patterns from data to help make conclusions if an activity is normal or abnormal. The system then develops a distance vector for the observed traffic and the profile. An alarm is raised by the system when the distance is great enough [15],[14], [36], [43]. These models are sub categorised into four: mean/standard deviation, multivariate, Markov process and operational model. Dorothy Denning [44] discusses model based on the hypothesis that security violation can be detected by monitoring a system's audit records for changes in pattern. The model includes profiles for representing the behaviour of subjects with respect to objects in terms of metrics and statistical models and rules for acquiring knowledge about this behaviour from audit records and detecting anomalous behaviour.

  a) Mean, standard deviation and any other form of correlations are known as moments in statistics [35], [45]. A moment is said to be anomalous when events fall either above or below a set interval. Decisions are made taking system change into account by altering statistical rule set for the system [35]. Its advantage over the operational model is its ability to detect attacks without having prior knowledge of the normal activities to set its limits. Rather it learns form observations to determine its normal activities [35]. It is a complex model that has more flexibility than the threshold model. It does not require prior knowledge to determine the normal behaviour to help set pre-defined threshold. Varying the mean and standard deviations slightly changes the computation by putting extra weights on the more recent values [35]. A.Ashfag *et al*[46] proposed a standard deviation normalised entropy of accuracy hybrid method of intrusion detection. Two real traffic dataset were used. The endpoint dataset comprising 14months of traffic traces on a diverse set of 13 endpoints. The dataset was reduced to 6weeks of traffic for testing and training purposes. The endpoints were infected with different malware attacks. The second was attack data dataset was obtained from two international network locations at Lawrence Berkeley National laboratory, USA on three distinct days. Using 9 prominent classifiers on the two datasets showed 3%-10% increase in detection rate and 40% decrease in false alarm rate over the existing classifiers can be achieved with the proposed hybrid technique.

  b) Multivariate models are similar to the mean and standard deviation model [35], [47]. The multivariate models are based on correlations between two or more metrics. They use multiple variables to predict possible outcomes. For example the number of CPU cycles can be compared to how long a

login session is completed [47]. Theoretically this model could have a fine distinction over one variable [47]. In an approach by W.Sha *et al*.[48], multivariate time series and high-order Markov chain were taken into account along the detailed design of training and testing algorithms. The models were evaluated using DARPA dataset. Observing the multi order Markov chain showed that the relative positions between results from models of different orders provide a new effective indication for anomalies. To improve sensitivity, a combining multiple sequences as a multivariate one into a simple model was applied and proved that the return of values of the system calls also play an important role in detection.

c) Markov process has two approaches: Markov chains and hidden Markov models. The Markov model is a set of finite states interconnected going through a stochastic process to determine the topology and capabilities of a model [36]. Each stage in the process depends on the outcome of the previous stage. The anomalies are detected [35] by comparison of the associated probability recorded for the process with a fixed threshold. This gives it an advantage of detecting unusual multiple occurrence of events [47]. The hidden Markov model assumes the system to be a Markov process where stochastic processes with finite states of possible outcomes are hidden [36]. Ye Nong [49] presented an anomaly technique using Markov chain model to detect intrusion. In this model, Markov chain was used to represent a temporal profile of normal behaviour in a computer and network system. The Markov chain model of the normal profile is learned from the historic data of the system's normal behaviour. The observed behaviour of the system is analysed to see if the Markov chain model of the normal behaviour supports the observed behaviour. A low probability of support indicates an anomalous behaviour meaning an intrusion. The technique was implemented on the Sun Solaris system and distinguished normal activities from attacks perfectly.

2.1.3 Signature Based Detection

Signature based detection defines a set of rules used to match the patterns in the network traffic. If a mismatch is detected it raises an alarm [50]. It has an advantage of being able to detect attacks giving a low false positive detection ratio [51] It has a drawback of being able to detect only attacks known to the database [19]. Signature based detection systems are programmed with distinct decision rules. The rules set for detection are coded in a straight forward manner to detect intrusion. They are programmed in four categories: state modelling, expert system, and string matching.

- State modelling is the encoding of attacks as a number of different states in a finite automaton. Each of these attacks has to be observed in the traffic profile to be considered as an intrusion. They occur in sub-classes as time series models [15]: the first is state transition which was proposed by Porras *et al*. which uses a state transition diagram to represent intrusion [52]. The approach in [52]models intrusion as a series of state transitions which are described as signature action and states descriptions. State diagrams were written to correspond to the states of an actual

computer system. The basis of a rule based expert system to detect intrusions was formed by these diagrams. The intrusions are in the form of a simple chain transitioning from start to end. The second is the petri-net where states form a petri-net. They form a tree structure where the transition states are not in any order [15][53].

- Expert systems contain a set of rules used to describe attacks scenarios known to the system. The given rules that describe the attack scenarios are often forward-chaining systems. A production based expert system tool has been used since they best handle systems with new events entering into the system. The size of the rule based increases as the execution speed increases since the rule will go through a longer list. It is however vulnerable to attacks not known to the set rules [2]. P.Zhisong *et al.*[54], presented an intrusion detection system model based on neural network and expert system. The aim of the experiment was to take advantage of classification abilities of neural network for probe and Dos attacks and expert based for U2R and R2L attacks. KDD Cup'99 dataset was employed in the experiment. Expert system was able to improve the detection accuracy using the detection rules: ALERT UDP ENET any <-> HNET 31337 MESSAGE:"BO access" DATA: |ce63 d1d2 16e7 13cf 3ca5 a586|".

- String matching is a process of knowledge acquisition just as Expert system but has a different approach in exploiting the knowledge [14]. It deals with matching the patterns in the audit event generated by the attack but not involved in the decision making process [47]. This technique has been used effectively commercially as an IDS [55], [56]. It is noted that not all signature based IDS can be represented by a simple pattern which gives it a limitation [57]. T.Sheu *et al.*[58]proposed an efficient string matching algorithm with compact memory as well as high worst-case performance. A magic number heuristic based on the Chinese remainder theorem was adopted. The algorithm significantly reduced the memory requirements without bringing complex processes. The latency of off-chip memory references was drastically reduced. It was concluded the algorithm gives a cost effective and an efficient IDS.

2.1.4 Discussion

A summary table of intrusion techniques, source of data, applications and data used in selected reviewed papers between 2014 and 2016 is shown in Table II.

Table II. Summary table of selected reviewed papers.

Authors/Year	Year Published	Intrusion Technique	Source of Data	Application	Data Set
Chetna and Harsh [59]	2014	Anomaly	NIDS	Cloud computing	Real Life
M. Ali Alheeti *et al.* [60]	2016	Anomaly	NIDS	Transport	Real Life
J. Hong *et al.* [61]	2014	Anomaly	NIDS	Substation	Real Life
R. Mohan *et al.* [62]	2015	Signature	HIDS/NIDS	Cloud computing	Real Life
S. Vasudeo *et al.* [63]	2015	Anomaly/ Signature	HIDS/NIDS	Computer systems	Real Life
W.Haider *et al.* [64]	2015	Anomaly	HIDS	Cloud computing	ADFA-LD and KDD98

H. Toumi et al. [65]	2015	Signature	NIDS	Cloud computing	Real Life
M. Guerroumi et al. [66]	2015	Signature	HIDS	Internet of things (IoT)	Real Life
N. Dipika et al. [67]	2015	Anomaly	NIDS	Information systems	KDD99 Cup
W.Haider et al. [68]	2015	Anomaly	HIDS	Cyber space	KDD98 and UNM
N.Aissa et al. [69]	2015	Anomaly	NIDS	Computer systems	KDD99
H. Omessaad [70]	2015	Signature	HIDS	Cloud computing	Real Life
X. Lin et al. [71]	2015	Signature	NIDS	Transport	Real Life
S. Banerjee et al. [72]	2015	Signature/ Anomaly	NIDS	Computer systems	Real Life
P. Satam [73]	2015	Anomaly	NIDS	Telecommunication	Real Life

Recent works have shown that works are still on-going using the Intrusion technique and source of data techniques. S.Vasudeo et al. [63] presented a hybrid of Signature and Anomaly with a hybrid of data source on a real life dataset. S.Banerjee et al.[72]also combined signature and anomaly technique on real life dataset. Other referenced works the table applied single techniques on real life datasets and KDD Cup datasets. The application of these techniques cut across all areas to determine their effectiveness in detecting intrusion.

3.0 MACHINE LEARNING TECHNIQUES

Anomaly detection systems are human-independent. They detect anomalies by revealing abnormal characteristics in a system over a period of time [44]. The effectiveness of this technique is its capabilities to differentiate between normal and abnormalities within a network.

Machine Learning (ML) can provide IDS methods to detect current, new and subtle attacks without extensive human-based training or intervention. It is defined as a set of methods that can automatically detect patterns to predict future data trends [74],[75]. Whilst a large number of machine learning techniques exist, the fundamental operation of all of them relies upon optimal feature selection. These features of are the metrics which will be used to detect patterns and trends. For example, one feature of a network is the packet size: machine learning techniques may monitor the packet size over time and generate distributions from which conclusions may be drawn regarding an intrusion. This section reviews the feature selection of machine learning IDS and classification of machine learning techniques used as IDS in Figure 3.

3.1 IDS feature selection

Machine learning classification involves two phases: the classification phase and training phase. The training phase learns the distribution of the features and during the classification phase the learned features is applied as a normal profile where any abnormality will be detected [76], [77]. A.K. Jain et al. [78] developed a model of statistical pattern recognition as shown in Fig.3. The test and training data are normalised in the processing unit as well as removing noise form the data. In the training phase, the feature extraction unit extracts a representation feature set from the processed training data which are used in training a classifier. In the classification phase, the trained classifier is applied to assign the test data to the selected features from the training phase [77].
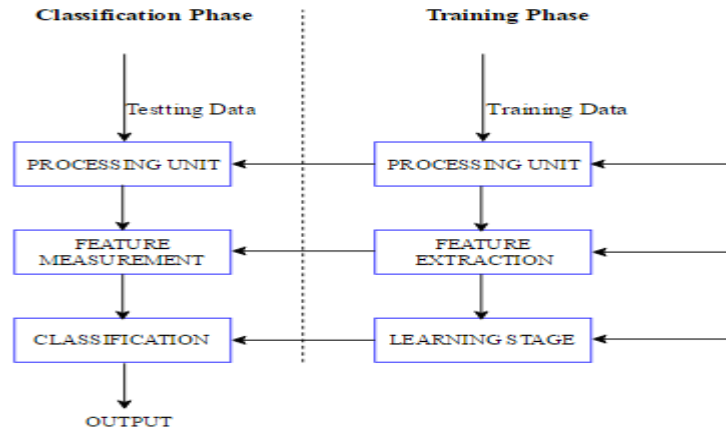
Figure 3 Model of Machine learning classification process

A high quality of training data is required to achieve the best performance of ML IDS. The training data thus contain both normal and abnormal patterns [76]. Features are the important information extracted form from raw data and are important in classification and detection which influence the effectiveness of ML IDS. In [79], C. Kruegel *et al*. created attacks containing system call sequences similar to normal system call. Kruegel *et a*l. addressed the issue by analysing on the system call arguments instead of finding the relations between sequences of actions [76]. Table III. Shows the features extracted by Kruegel *et al*. for four different arguments.

Table III. Extracted features by Kruegel *et al*. [76] for four different arguments

No.	Feature Description
1	String length
2	String character
3	Structural inference
4	Token finder

In [80], V. Mahoney *et al.* extracted features by analysing each header-field of network packets and flows. The proposed approach det6ected anomalies in a network based on 33 fields of IP, UDP, TCP, ICMP and Ethernet protocols. Table IV shows some of the extracted features.

Table IV. Extracted features based on analysis of each header field

No.	Feature Description	No.	Feature Description
1	IP Source	10	IP protocol
2	IP Destination	11	IP Length
3	TCP Source Port	12	IP Type of Service
4	TCP Destination Port	13	IP Header Length
5	UDP Source Port	14	IP Time to Live
6	UDP Destination Port	15	Ether Source
7	UDP Length	16	Ethernet Size

10

8	ICMP Code	17	Ethernet Destination
9	ICMP Type	18	Ethernet Protocol

In [81], W. Lee *et al.* extracted features from TCP/IP connections. Experiment was conducted on 1998 DARPA dataset where a set of basic features from domain knowledge was extracted. Most of the basic features could only be extracted after the TCP connection was terminated leading to a delay in the detection phase [76]. The 1998 DARPA dataset is categorised into five. These are NORMAL and four different types of attack. The attacks are: Dos attacks, probe attack, User to root (U2R) attack and Remote to local (R2L) attacks. Dos attack which denies or prevents legitimate users resource on a network or system by introducing useless or unwanted traffic. Probe is an attack which scans through a computer network to make a profile of information for future attacks. U2R is attacks use a local account from a remote machine to gain access to the targets system due to vulnerabilities in its operating system. R2L attacks are initiated to gain unauthorised access to root privileges from outside. Table V. illustrates the basic features of individual TCP connections by W. Lee *et al.*

Table V. Basic features extracted from individual TCP connections

No.	Feature Name	Feature Description	Type
1	Duration	Length of connection in seconds	continuous
2	protocol_type	Type of the protocol used, e.g.. up,tcp, etc.	discrete
3	Service	Network service on the destination, e.g.http, telnet, etc.	discrete
4	scr_bytes	Number of data bytes from source to destination	continuous
5	dst_bytes	Number of data bytes form destination to source	continuous
6	Flag	Normal or error status of network	discrete
7	Land	1 if connection is from/to same host/port; 0 otherwise	discrete
8	wrong_fragment	Number of wrong fragments	continuous
9	Urgent	Number of wrong fragments	continuous

Table VI. Traffic features extracted from a 2second time window

No.	Feature Name	Feature Description	Type
10	Count	Number of connections to the same host as the current connection in the past 2seconds	continuous
11	serror_rate	Percentage of connections having "SYN" error	continuous
12	rerror_rate	Percentage of connections having "REJ" error	continuous
13	same_srv_rate	Percentage of connections having same service	continuous
14	diff_srv_rate	Percentage of connections having different service	continuous

15	srv_count	Number of connections to the same service as the current 1connection in the past 2seconds	continuous
16	srv_serror_rate	Percentage of connections having "SYN" errors	continuous
17	srv_rerror_rate	Percentage of connections having "REJ" errors	continuous
18	srv_diff_host_rate	Percentage of connections to different host	

Table VII. Traffic features extracted from a window of 100 connections

No	Feature Name	Feature Description	Type
19	dst_host_count	Count for destination host	continuous
20	dst_host_serror_rate	Percentage of connections having "SYN" errors	continuous
21	dst_host_rerror_rate	Percentage of connections having "REJ" errors	continuous
22	dst_host_same_srv_rate	Percentage of connections having same service	continuous
23	dst_host_diff_srv_rate	Percentage of connections having different service	continuous
24	dst_host_srv_count	Number of connections to the same destination port	continuous
25	dst_host_srv_serror_rate	Percentage of connections having "SYN" errors	continuous
26	dst_host_srv_rerror_rate	Percentage of connections having "REJ" errors	continuous
27	dst_host_srv_diff_host _rate	Percentage of connections to different host	continuous
28	dst_host_same_src_port _rate	Percentage of connections to the same source port	continuous

Table VIII. Traffic features extracted within a connection suggested by domain knowledge

No.	Feature Name	Feature Description	Type
29	Hot	Number of "hot" indicators	continuous
30	num_failed_logins	Number of failed login attempts	continuous
31	logged_in	1 if successfully login; 0 otherwise	discrete
32	num_compromised	Number of "compromised" conditions	continuous
33	root_shell	1 if root shell is obtained; 0 otherwise	discrete
34	su_attempted	1 if "su root" command attempted; 0 otherwise	discrete
35	num_root	Number of "root" accesses	continuous

36	num_file_creations	Number of file creation operations	continuous
37	num_shells	Number of shell prompts	continuous
38	num_access_files	Number of operations in access control files	continuous
39	num_outbound_cmd	Number of outbound commands in an ftp session	continuous
40	is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	discrete
41	is_guest_login	1 if the login is a "guest" login; 0 otherwise	discrete

W. Lee *et al.* discovered many attacks such as R2L and U2R were in the payloads of packets. This prompted a proposal to combine features extracted from the payload with domain knowledge. They called the features "time based traffic" features for connection records. Features 10 to 18 in table VI are extracted using a 2second time window with 11 to 15 having same host connection and 16 to 18 having same service connection, 19 to 28 in Table VII are extracted from a window of 100 connections with 20 to 24 same host connection and 25 to 28 same service connection. 29 to 41 in Table VIII are features extracted from connections suggested by domain knowledge [77]. Both tables provide a detailed insight mapping the most current and meaningful features for machine learning intrusion detection systems.

3.2 Classification of Machine Learning Techniques

This section gives an extensive classification of ML techniques. Each of the techniques is described in detail and how these techniques have been applied as IDS. Figure 3.1 shows the classification of Machine learning IDS.

Figure 3.1 ML based techniques

- **Bayesian Networks** are graphical modelling tools used to model the probability of variables of interest. They are directed acrylic graphs where each node represents a discrete random variable of interest. Each node contains the states of the random variable in tabular form representing the conditional probability table (CPT) which specifies conditional probabilities of the domain variable with other connected variables. The CPT of each node contains probabilities of the node being in a specific state given the domain variable states. The existence of the relationship between the nodes of the domain variable and connected variables in a Bayesian networks show the direction of causality between other connected variables. Thus the connected variables are causally dependent on the ones represented by the domain variable. Given a set of discrete random variable $X = \{x_1, x_2 \dots x_n\}$, the joint probability of the variable can be computed based on Bayes Rule as:

$$P(x_1, x_1 \dots x_n) = \prod_{t=1}^{n} P(x_i \backslash p_a(x_i)) \qquad (1)$$

where $p_a(x_i)$ represents the specific values of the variables in the domain variable node of x_t [82], [83]. This technique has generally been used as an intrusion detection system.

A. Onik *et al.*[84] Conducted an experiment using Bayesian networks on NSL-KDD dataset containing 25,192 records with 41 features. Apart from the normal class label it contained 4 more class of attacks known as Dos, U2R, R2L and probe attacks. The filter approach of feature selection was used to reduce the dataset features from 41 to 16 important features. Bayesian model was built and proved to predict attacks with superior overall performance accuracy rate of 97.27% keeping the false positive rate at a lower rate of 0.008. The model as compared to Naïve Bayes, K-means clustering, decision stamp and RBF network recorded 84.86%, 80.75%, 83.31% and 91.03% respectively in terms of accuracy.

An experiment by M. Bode *et al.*[85] Analysed the network traffic in a cyber situation with Bayesian network classifier on the KDD Cup'99 dataset with 490,021 records. The data set was made up of 4 types of attacks (DoS, U2R, R2L and probe). In this experiment, they adopted the risk matrix to analyse the risk zone of the attacks. The risk analysis adopted showed DoS was most frequent attack in occurrences. The results showed Bayesian network classifier is a suitable model resulting in same performance level classifying the DoS attacks as association rule mining. Bayesian network classifier outperformed Genetic Algorithm in classifying probe and U2R attacks and classified Dos equally.

- **Genetic Algorithm (GA)** is an adaptive search method in a class of evolutional computation using techniques inspired from convolutional biological process. The principle is based on a stochastic global [25] search method initialising with a random generation of chromosomes. The chromosomes are called population. They evolve through selection, crossover and mutation as shown in Figure 3.2. Each chromosome represents a problem to be solved and encoded as strings. The positions of the chromosomes are commonly represented as binary (0, 1) or as a list of integers. These positions sometimes referred to as genes keep changing at each initialisation. The solution created during each generation is based on an evaluation function. The selection is thus based on the chromosome fitness level [86], [87].
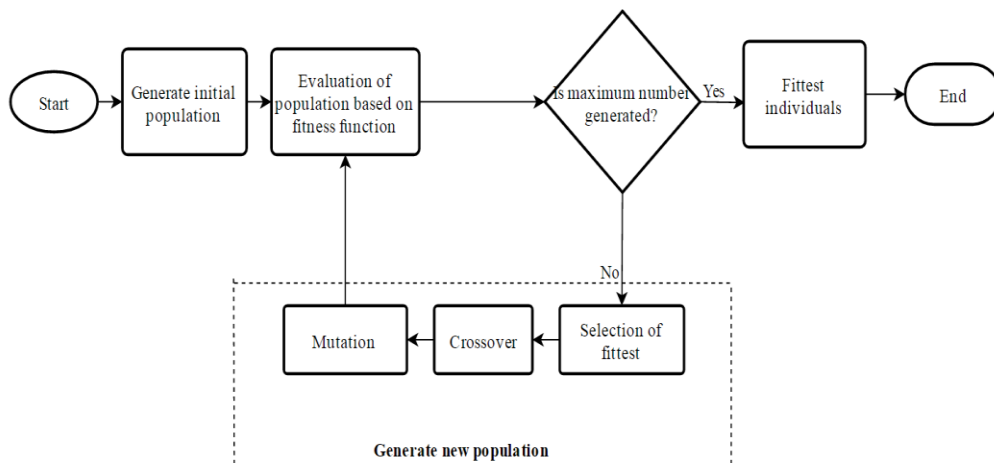

Figure 3.2 Flow chart of Genetic Algorithm

Tao Xia *et al.* [88] developed a hybrid method based information theory and GA. Information theory was used to filter out the most important features out of 41 features in KDD'99 dataset with 494021 records. Linear rule was used initially to classify normal and abnormal before GA to obtain the appropriate classification. In the detection of Dos, U2R, R2L and probe attacks, the information theory and GA hybrid method recorded 99.33%, 63.64%, 5.86% and 93.95% as detection rates respectively. The detection rates were compared to Ctree and C5. Ctree recorded 98.91%, 88.13%, 7.41% and 50.35% whilst C5 recorded 97.1%, 13.2%, 8.4% and 83.3% respectively for the attacks.

In layered approach for intrusion detection systems based GA, M. Padmadas *et al.*[89] Proposed a method to overcome the weakness in a single layer intrusion detection system. The layered approach is based on GA with the four layers corresponding to four groups of attacks (probe, DoS, U2R and R2L). Each layer is trained separately with a number of features where the layer acts as a filter to block any malicious activity. In layered approach there is no mathematical approach to calculate the filter parameters for the attacks. This paper presented GA approach in calculating the filter parameters making the system more secure. The model efficiently detected R2L attack and recorded an accuracy of 90% in detection.

- **Support Vector Machines (SVM)** is a machine learning algorithm that learns to classify data using points labelled training examples falling into one or two classes. The SVM algorithm builds a model that can predict if a new example falls into one category or the other [90],[25], [91] . Figure 3.3 shows a hyperplane defined by (w, b), where w is a weight and b bias constructed in a finite space of the training sample N with points:

$$\{(x_1, y_1)(x_2, y_2 \ldots (x_N, y_N))\} \tag{2}$$

Where $x_i \in R^d$ and $y_i \in \{1, -1\}$. This is conducted since in general the larger the margin, the lower the generalisation error of the classifier [92].
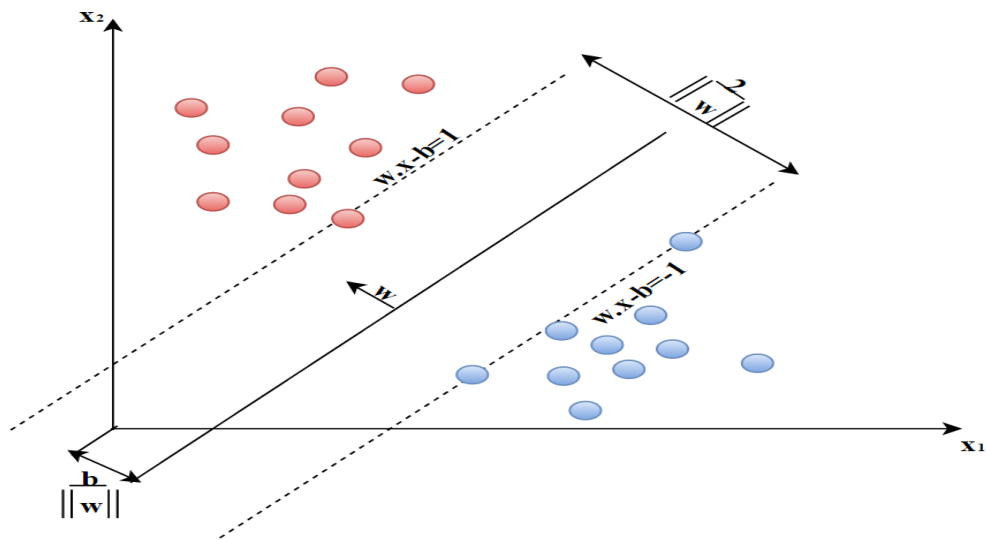

Figure 3.3 Maximum-margin hyper plane and margins for an SVM trained with samples from two classes.

B. Senthilnayaki *et al.*[93] Used GA pre-processed KDD Cup 99 dataset in a pre-processing module for data reduction since it was complex to process the dataset with all 41 features. GA was used to select 10 features out of 41 features present in the KDD Cup 99 dataset and applied SVM for classification. The experiment was carried out with 100,000 records from the dataset out of which 95% was used as training data and remaining 10% as test data. The classification process continued till a 10 fold cross validation was done for results verification. The SVM classified four different attacks (DoS, probe, U2R, R2L attacks). The performance of SVM classifier was compared with four other classifiers as shown in table IX.

Table IX. SVM performance analysis SVM, GMM, Naive, MLP and linear algorithm

Classified Algorithm	Accuracy% (41 feature)	Accuracy% (10 features)
Gmm	30.5	45.07
Naive	45.17	55.75
MLP	55.24	67.87
Linear	65.47	76.75
SVM	82.45	95.26

Another experiment by L.Teng *et al*. [94]proposed a novel method integrating PCA and SVM by optimising the kernel parameters using automatic parameter selection technique. The experiment was performed on KDD Cup'99 dataset containing five categories of traffic (normal, DoS attack, R2L attack, U2R attack and probe attack). Each network record had 41 features of which 7 were discrete and 34 continuous features. C parameter for RBF kernel of SVM was optimised by the proposed automatic parameter reduction along with cross validation to reduce the training and testing time to give better accuracy in detecting attacks.

- **K-Nearest Neighbour (K-NN)** is a fundamental technique for sample classification. This technique is known to be non-parametric and highly efficient in classification [95]. It evaluates the class labels of the test samples [96] based on the majority of test sample neighbours. The parameter k is determined by the user. Based on the test sample, k numbers of training points are determined by taking the closest distance to the test sample. The prediction of the test sample is the k nearest neighbours [97].
A hybrid method of intrusion detection system by Y.Canbay *et al*.[96]proposed combination of K-NN and GA algorithm. They tested the hybrid algorithm on KDD Cup 99 dataset labelled out of five classes; normal, probe attack, Dos attack, R2L attack and U2R attack. The dataset was reduced to 1000,2000,3000,4000 and 5000 records respectively with 19 features. GA was used to select the k nearest neighbour for K-NN classifier. Three experiments were performed to conclude on the hybrid method. Each experiment used 10 fold cross validation with different k values. The values in in terms of accuracy were compared with the conventional K-NN. The proposed hybrid method proved better than conventional K-NN in all values of k used in the experiment.

On the same dataset Q.Zeng *et al*. [98]compared the performance of K-NN and SVM model and RIPPER method in detecting attacks. The multi attribute decision was adopted in this experiment. In the classification of an unknown document vector X, k–nearest neighbour algorithm ranks the document's neighbour among the training document vectors and uses the class labels of the k most similar neighbours to predict the class of the new document. The similarity in each neighbour to X is used to determine the classes of the neighbour where the similarity is measured by the Euclidean distance between two document vectors. With this adoption they categorised each new program behaviour in the dataset into either normal or attack class. Each system call was treated as a word and each process as a document. k was varied between 15 and 35 till an optimal value of 19 found. The classification was performed with K-NN and SVM model. The Hit rate was compared to the RIPPER method. The results showed 97.26% accuracy rate and 6.03% false alarm rate for K-NN and SVM model whilst the RIPPER method gave 87.26% accuracy rate and 8.6% false positive rate.

- **Decision Tree (DT)** algorithm learns and models a data set in classification problems. It classifies new data set according to what it has learnt from previous data set [99]. It uses a well-defined criterion in the selection of best features of each node tree during their construction. A decision tree model has a root node linking to different nodes as attribute data deciding the path for each node [100]. Decisions are made by comparison of previous data and marked as leaves [101]. A common decision tree approach is the C4.5 algorithm.

In network intrusion detection system using decision tree (J48) by S.Sahu *et al*. [102] a labelled data set called Kyoto 2006+ was used. The data consist of 24 features, 14 of which was extracted form KDD Cup 99 dataset and an additional ten important features. The Perl language was used to extract 15 features from the dataset for the experiment. The sample dataset contained 134665 records; 44257 normal, 86649 known attacks and 3759 unknown attacks. Decision tree (J48) built using WEKA 3.6.10 tool was used to classify normal, known attacks and unknown attacks in the network packets. The results showed the decision tree generated classified 97.23% correctly and 2.67% incorrectly. The simulation results showed decision tree can classify unknown attacks as well.

T.Komviriyavut *et al*.[101]presented two intrusion detection techniques which were decision tree (C4.5) and RIPPER rules to test an online dataset (RLD09 dataset). RLD09 dataset was collected from actual environment and refined to have 13 features. The dataset is categorised into three; normal, DoS attack and Probe attack. The experimental data set had 3,000 records of three types of unknown probe attacks each with 1,000 attack records. Initial experiment with known attacks on dataset showed a total detection rate of about 98% for both decision tree and RIPPER rule.

A second experiment on the same data set with three unknown probe attacks (advance port scan, Xmas tree and ACK scan) was performed. The decision tree maintained its detection rate of 98% whilst the RIPPER rule degraded to about 50%in average.

- **Fuzzy logic (FL)** concept is derived [103] from the fuzzy sets theory which deals with approximately reasoning with uncertainty and imprecision [104] by human being. The features this technique which handles real life uncertainty makes it

attractive for anomaly detection. Intrusion detection involves the classification of a normal class and an abnormal class. The well-defined nature of the two classes makes this computation paradigm a helpful one.

A.Toosi *et al.*[105]introduced a new approach using different soft computing techniques into classification system to classify abnormal behaviour form normal depending on the attack type. This work investigated neuro-fuzzy networks, fuzzy inference and GA on KDD Cup 99 10% dataset. The dataset had a distribution of normal and four different attacks (probe, DoS, R2L and U2R). A set of parallel neuro-fuzzy classifiers were used initially for classification. The fuzzy inference system was based on the output of the neuro-fuzzy classifier determining normal or abnormal activity. The best results were attained by optimising the structure of the fuzzy decision tree with GA.

- **Clustering** is the division of data with no apparent distinguished differences into groups by functions of two or more parameters. Each group is called a cluster and have no similarity. The greater the dissimilarity the better the grouping. Their class labels are always unknown. K-means is an example of clustering algorithm used as IDS [106],[107]. K-means algorithm makes use of the Euclidean distance to calculate the data and the cluster centre. The aim is to have a minimum distance within a cluster and achieve a maximum distance between clusters by the minimisation of an objective function [108].

  Li Jun Tao *et al.*[109]proposed a k-means clustering with dynamic adjustable number of cluster. They introduced an algorithm that uses an improved Euclidean distance formula to calculate the distance between the data and cluster centre by automatically adjusting the number of cluster when the distance is more than the threshold. This experiment was performed on KDD Cup'99 dataset containing four attacks (DoS, Probe, R2L and U2R). The total number of dataset records was 72471. The improved k-means algorithm recorded 77 clusters with a detection rate of 90% and false positive rate of 15%. The results compared to the traditional k-means were 108 clusters, detection rate of 85% and false positive rate of 42%.

3.3 Analytical Comparison of ML Techniques

The various ML techniques discussed in the above section are different in their capabilities to classify attacks. Table X. analyses the advantages and disadvantages in the performance of ML techniques.

Table X. Advantages and disadvantages of ML techniques

ML Technique	Advantages	Disadvantages
Bayesian Network	The graphical representation gives it an advantage of breaking complex problems into different smaller models.	Slow in classifying data sets with many features.
Genetic Algorithm	-Uses a technique which is inspired by convolutional biological process. -It has the capability of solving optimisation problems during classification.	Gets stack in local optima (overfitting)
Support	-The algorithm is simple to analyse	-Selection of kernel function is not straight

Vector Machine	mathematically. -All computations are performed in space using kernels giving it an edge to be used practically.	forward. -Slow in training and requires more memory space.	
K-Nearest Neighbour	Easy to implement and can solve multi-class problems	-Slow in training and requires large memory space. -It is computationally complex because to classify a test sample involve the consideration of all training samples.	
Decision tree	-It has a unique structure therefore easy to interpret. -It has no limitation in handling high dimensional data sets.	-If trees are not pruned back it causes overfitting. -Type of data must be considered when constructing tree. (i.e. Categorical or numerical)	
Fuzzy Logic	-It is based on human reasoning concepts which are not precise. -It gives a representation of uncertainty.	-It's construction has a high level of generality there by high consumption of resource.	
K-means Algorithm	Simple to implement and effective	-The outcome of clustering depends on how cluster centres are initialised to specify k value. -The algorithm works for only numerical data.	

3.4 ML Hybrid Techniques

ML Intrusion detection systems have generally been used to detect attacks but in recent times, using two or more different techniques to form a hybrid has improved the overall performance. Table XI. Shows accuracy of detection, type of attacks and data set for selected reviewed ML IDS papers from 2011-2016.

Table XI. ML IDS accuracy of detection and Data set used

Authors(s)/Year	ML Technique	Attack Type(s)	Data Set	Accuracy
S.Chordia *et al.* [107] **2015**	K-means +KNN+DT	R2L,U2R,DoS and probe	KDDCup99	96.55%
P. Jongsuebsuk *et al.* [110] **2013**	FL+GA	Dos and Probe	Real Life	97%
B.Senthilnayaki *et al.* [93] **2015**	GA+SVM	Dos Probe U2R R2L	KDDCup99	99.15% 99.08% 97.03% 96.50%
B. Masduki *et al.* [7] **2015**	SVM	R2L	KDDCup99	96.08%
A.Enache *et al.* [111] **2015**	SVM+BAT Algorithm	Malicious	NL-KDD	99.38%
S. Akbar *et al.* [112] **2012**	GA	DoS,R2L,U2R and Probe	KDDCup99	92.6%
A. Aziz *et al.* [113] **2013**	DT	Dos Probe	NSL-KDD	82% 64%
Chao Lin *et al.* [114] **2015**	Cluster centre+ K-NN	Probe Dos	KDDCup99 (6 dimensional data set)	99.98% 99.99%
A.Aburomman *et*	SVM+K-NN	DoS,R2L,U2R and	KDDCup99	87.44-91.69%

al. [115] 2016		Probe		
Shi-Jin Horng *et al.* [116] **2011**	SVM + hierarchical clustering algorithm	DoS,R2L,U2R and Probe	KDDCup99	95.72%
E. Hodo *et al.*[117] **2016**	ANN	DDoS/Dos	Real Life	99.4%

As demonstrated in Table XI, the majority of Intrusion Detection Systems using machine learning have tested their work against the KDDCup99 dataset or the NSL-KDD, in contrast only two recent studies tested their machine learning systems against real network data.

3.5 Binary Classification metrics

The effectiveness of prediction by the ML algorithm which is either 1 or 0 is based on confusion matrix prediction outcome as shown Table IX. The outcomes are True Negative (TN), True Positive (TP), False Positive (FP) and False Negative (FN) [118], [119].

Table IX. Confusion Matrix

		Predicted Class	
		Negative Class(Normal)	**Positive Class(Attack)**
Actual Class	**Negative Class(Normal)**	**True Negative(TN)**	**False Positive(FP)**
	Positive Class(Attack)	**False Negative(FN)**	**True Positive(TP)**

- True Negative (TN): a measure of the number of normal events rightly classified normal.
- True Positive (TP): a measure of attacks classified rightly as attack.
- False Positive (FP): a measure of normal events misclassified as attacks.
- False Negative (FN): a measure of attacks misclassified as normal.

The following are the basic metrics used to calculate the performance of ML IDS:

True negative rate (Specifity) $= \frac{TP}{FP+TN}$ (3)

True positive rate (Sensitivity) $= \frac{TP}{TP+FN}$ (4)

False positive rate (Fallout) $= \frac{TP}{TP+FN} = 1 - Specifity$ (5)

False negative rate (Miss Rate) $= \frac{FN}{FN+TP} = 1 - Sensitivity$ (6)

Precision $= \frac{TP}{TP+FP}$ (7)

$$\text{Recall} = \frac{TP}{TP+FN} \tag{8}$$

$$\text{Overall accuracy} = \frac{TP+TN}{TN+TP+FN+FP} \tag{9}$$

A commonly used ML IDS metric is detection rate. This is defined as the number of data examples correctly classified divided by the test examples.

4.0 ARTIFICIAL NEURAL NETWORK AND DEEP NETWORKS IDS

This section reviews artificial neural network (ANN) and Deep learning which use the computational intelligence approach to detect attacks.
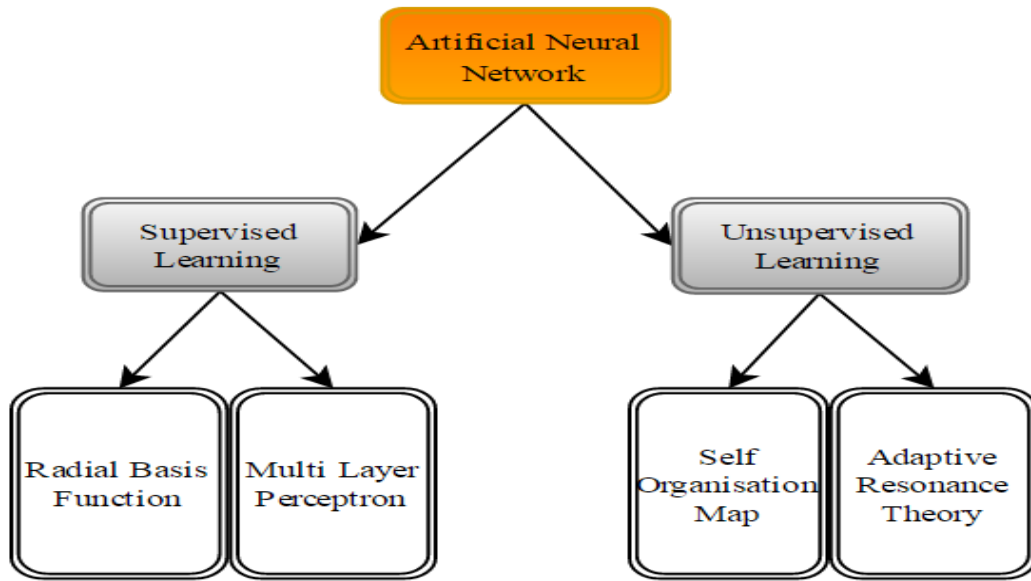
4.1 Artificial Neural Network

Figure 4 Classification of Artificial Neural Network

Artificial neural network consist of information processing elements known to mimic neurons of the brain. ANN is categorised into supervised and unsupervised learning. Figure 4 shows the types of ANN.

4.1.1 Supervised Learning

In supervised learning, the neural network is provided with a labelled training set which learns a mapping from inputs x to outputs y given a labelled set of inputs-output pairs

$$d = \{(x_i, y_i)\}_{i=1}^{N} \tag{10}$$

Where d is called the training set and N is the number of training examples. It is assumed that y_i is a categorical variable from some infinite set $y_i \in \{1 \dots C\}$ [120]. Two types of supervised learning algorithms are used to train a neural network for intrusion detection.

- Multilayer Perceptron (MLP) is a feedforward neural network. The structure of MLP consists of one or more nodes as the inner layer between the input and output nodes as shown in Figure 4.1. The most common technique used to train the MLP neural

network is the Back Propagation hence the name MLP-BP. The construction of the MLP-BP neural network is by putting layers of non-linear elements to form complex hypotheses. The more stages that are added (nodes) the more advance the hypotheses. Each node takes an element of a feature vector. The output nodes give an output of two classes (normal and attack). The interconnection between the nodes is associated with scalar weights with an initial weight assigned to the connection. During training, the weights are adjusted. Evaluating the hypotheses is done by setting the input modes in a feed-back process and the values are propagated through the network to the output. At this stage the gradient descents is used so as to push the error in the output node back through the network by a back propagation process in order to estimate the error in the hidden nodes. The gradient of the cost − function can thus be calculated [99], [121].
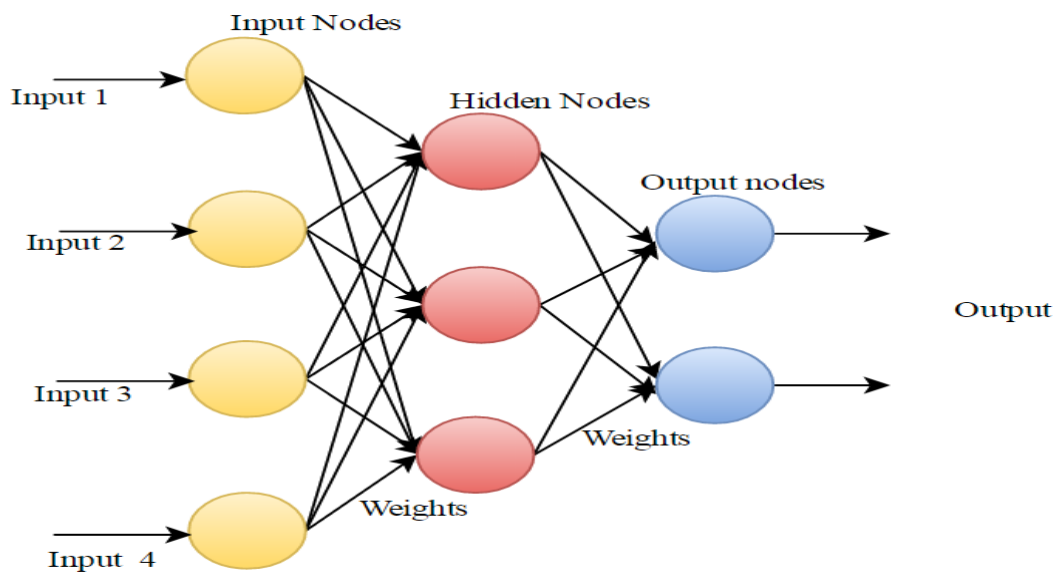
Figure 4.1 Three layer neural network

In [122], P. Barapatre *et al.*[122]experimented with neural network having an input layer, one hidden layer and one output layer. In this experiment, the input node had 41 features from KDD Cup'99 dataset. The output node was to classify normal or attack present in dataset. If the output node is '0' then it is labelled "normal" and if '1' it is labelled "attack". The learning rate was initially set to 0.1 and the system retrained to reduce learning rate. A sigmoid activation function was used and values corresponding to the weights randomly initialised between -1 and -. The system was trained for each category of attacks (DoS, Probe, U2R and R2L attacks) to determine the performance of the algorithm on individual attacks. Table IIX shows the experimental results. It was concluded MLP-BP neural network detected DoS and Probe attacks more accurately than U2R attacks. It observed an increase in rate of classification as rate of learning decreases giving rise to a slow convergence. Also reducing the rate of learning and the sum squared error (SSE) values and re-training the network showed an improvement in rate of detection.

Table IIX Detection rate comparison for different attack types using MLP

Attack type	Detection rate %	False positive rate %	False negative rate %
DoS	99.75	4.78	0.24
Probe	98.16	1.33	1.83
U2R	87.09	31.03	12.9
DoS-Probe	99.33	18.5	0.66
R2L	98.99	9	1.01
Overall	81.96	8.51	18.03

- Radial Basis function (RBF) is another feed forward neural network. It classifies by taking a measurement of the distance between the inputs and the centre of hidden neurons [118]. Figure 4.2 is an RBF architecture showing the input nodes, one hidden nodes and output. Each RBF has different parameters with an input vector. The network output is thus a linear combination of the radial basis function's output. The input and hidden nodes weights are always 1 since the transfer function of the network is a Radial basic function. This allows an adjustment on the weight between the hidden nodes and the output [123].
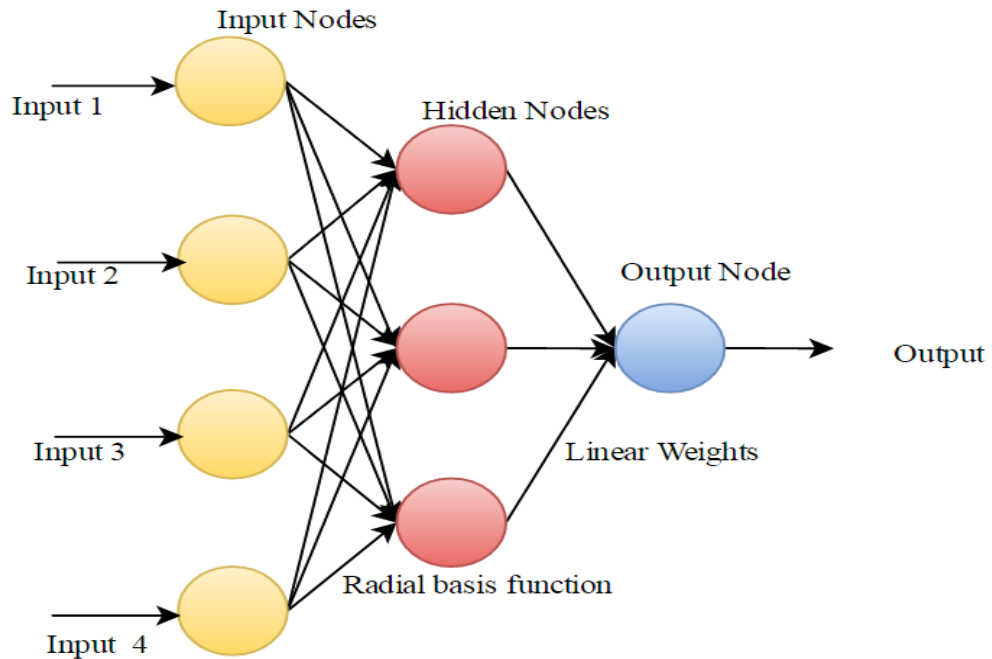
Figure 4.2 Radial basis function architecture

C.Zhang *et al*.[124] compared the performance of RBF and MLP-BP in detection detecting four different types of attacks on KDD Cup'99 dataset. Both training and testing data contained 1000 records which were selected from the dataset. In this experiment each sample was unique with 34 numerical features and 7 symbolic features. The symbolic features were converted to ASCII numbers before being used as training or testing data. The results showed RBF achieved better performance than MLP-BP with a detection rate of 99.2% and false positive rate of 1.2%. MLP-BP showed a detection rate of 93.7% and false positive rate of 7.2%.

Ju Jiang *et al*. [125]applied RBF and Back propagation algorithm (BPL) to both misuse and anomaly detection. The experiment was performed on KDD Cup '99 dataset containing 4,900,000 records categorised into 5 (normal, probe attacks, DoS attacks, R2L attacks and U2R attacks). The misuse detection network had 41 input features, 4 hidden nodes and 4 output nodes representing normal probe, DoS and R2L classes respectively. The anomaly network also had 41input features, 1 hidden node and 1 output node to classify normal and attacks behaviour. The experimental results showed in misuse detection, RBF based IDS performed similarly to BPL based IDS. However RBF used a shorter time in training as compared to BPL based IDS and needed to adjust its decision thresholds. In anomaly detection, the BPL based IDS had to adjust itself output threshold manually according to the characteristics of the training dataset to achieve best performance. In anomaly detection RBF based IDS out performed BPL based IDs.

Bi Jing *et al*. [123] compared RBF and MLP-BP using a processed KDD Cup '99 dataset by converting al strings to numeric , reducing the dimension of the dataset and determining the rational value domain. The features were reduced from 41 to 31 to train both RBF and MLP-BP. The RBF structure used had one hidden layer and output neuron being the weighted sum of all the output items of the hidden layer. The simulation results showed RBF network is better than MLP-BP in its property of having a more regular out with shorter training time and better accuracy in attack detection.

4.1.2 Unsupervised learning

In unsupervised learning, the neural network is only provided with input data without conceptualising the output: it discovers patterns within the data autonomously. The data yet to be discovered is called unlabelled data [120]. Two typical unsupervised learning are Self-Organisation Maps (SOMs) and Adaptive Resonance theory (ART).

- Self-Organisation Maps

  SOMs transform the input of a network into two dimensional feature maps based on the topological properties of SOMs. The computation of feature maps is by Kohonen unsupervised learning. The two dimensional feature maps are neurons represented by coloured squares showing the weights corresponding to each neuron. The inputs are grouped based on their similarity. The more features mapped, the bigger the coloured square. The quality is determined by the back ground colouring of the clusters. Anomaly events can thus be identified by analysing the normal and abnormal events from the mapping [118], [126].

  SOMs are widely used anomaly detection systems. P. Lichodajewski *et al*. in [127] applied SOMs as a Host based intrusion detection system. In this work SOM was trained with an explicit coding of data and gave a clear clustering of abnormal behaviours.

  In [128], H. Gunes Kayacik *et al*. investigated and demonstrated with hierarchical SOM architecture with two basic feature sets , one limited to 6 basic features with the others containing all 41-features. The results gave a false positive rate of 1.38% and detection rate of 90.4%.

V. Kumar *et al* in [129] presented a unified framework on SOM. Their approach detected attacks on a mobile ad-hoc network (MANET) using different parameters. Their experimental results were found to be better than other neural network approaches in terms of detection rate and false alarm rate.

- Adaptive resonance Theory (ART)

  ART as a basic theory is an unsupervised learning model but as a hybrid it performs supervised learning. It functions as a pattern recognition and predictive tool. As an unsupervised learning model, ART-1, ART-2, ART-3 and fuzzy Art makes the list. The supervised models are made up of ARTMAP, Fuzzy ARTMAP and Gaussian ARTMAP [118]. The model compares an input vector to a single neuron's weight (weight vector) [118], [126].

  In [130], M. Chauhan *et al*. presented a novel technique by uploading the weights of the network based on sign of an evaluation function [131]. An appropriate evaluation function was selected by utilizing the probabilities of various states. It concludes that if the weights in a modified ART network are oscillating, it implies an intrusion and thus the ART network has the capability of detecting any intrusion.

  A hybrid method by P. Somwang *et al.* in [132] used principal component analysis (PCA) and fuzzy adaptive resonance theory to identify different attacks on KDD Cup '99 data set. The results showed a high performance of detection rate of 96.13% and a false alarm rate of 3.86% of anomaly intrusion detection.

4.1.3 Summary

I. Ahmad *et al*. in [133] evaluated SOMs, ART and MLP-BP in terms of main criteria and sub-criteria using analytic Hierarchy. Evaluation based on main criteria investigated less overhead, maturity, competency, performance and suitability. Sub-criteria investigated cost effective, time saving, detection rate, minimum false positive, minimum false negative, handling varied intrusion and handling coordinated intrusion. Different radar graphs with distinct colours were used for the analysis.

Analysing MLP-BP and SOM showed MLP-BP had a better detection rate, minimum false positive, minimum false negative, time saving and cost effective. In the case of less overhead and capability of handling coordinated and varied intrusion, MLP-BP was not as good as SOM. Analysing MLP-BP and ART showed the same results.

A comparison of ART and SOM showed better results for SOM in terms of less overhead and handling coordinated intrusion. In another scenario ART was better than SOM in terms of detection rate, minimum false negative, maturity, time saving and cost effective.

In conclusion, a hybrid ANN approach was found to be the most suitable intrusion detection system in terms of detection rate, false positive, false negative, cost and time saving.

4.2 Deep Networks

This section focuses on networks that look like the multilayer perceptron (MLP) but have a different architecture. The difference between the MLP and the deep networks is their training procedures. This is a class of ML techniques where classification is conducted by training data with many layers in hierarchical networks with unsupervised learning. Deep

networks are inspired by the architecture depth of the brain. In 2006, Hinton et al. from the university of Toronto came up with Deep Belief Network (DBN) [134]. They trained data with an algorithm that greedily trains layer by layer using unsupervised learning for each layer of Restricted Boltzmann Machine (RBM) [135]. After this discovery by Hinton et al. other deep networks have been introduced using the same principle have been successful in classification task [136]. Deep networks IDS can be classified based on how the architectures and techniques are being used. Figure 4.3 shows a classification of Deep networks IDS.
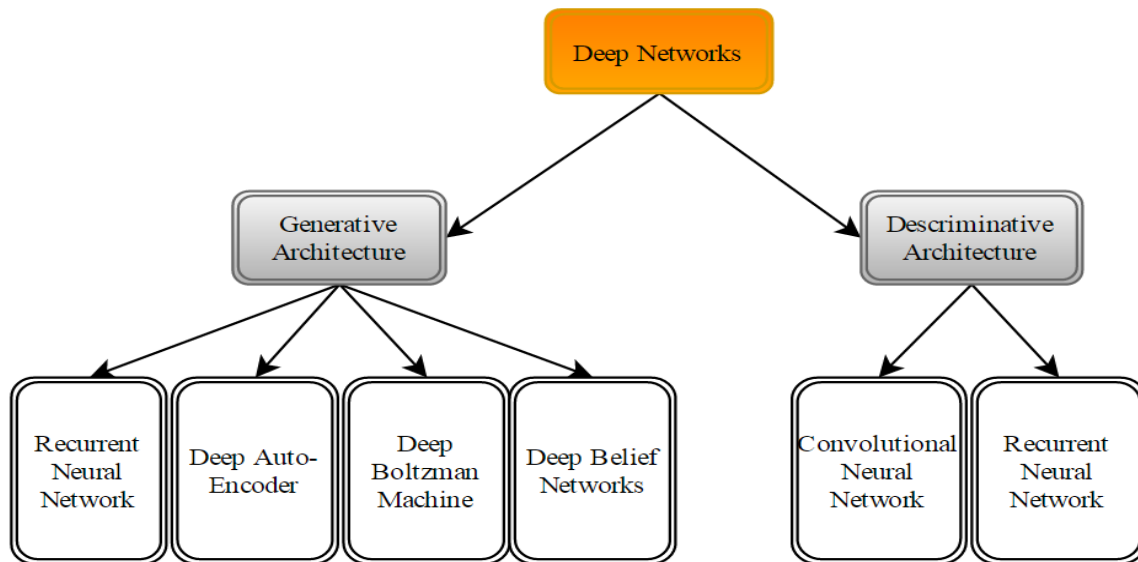
Figure 4.3 Classification of deep learning IDS

4.2.1 Generative Architecture

Generative models are also referred to as graphical model because they depict independence/dependence for distribution. They are visualised as graphs that have nodes representing random variables and arcs show the relationship between random variables that can have millions of parameters to graphically represent the given system [136], [137]. The joint statistical distribution of the variables can be written as products of the nodes and their associated variables [136], [138]. The graphical models have hidden variables that cannot be observed. Generative models are associated with supervised learning since their training does not depend on the labels of the data. For classification purposes the models goes through a pre-training stage (unsupervised learning). During this process, each of the lower layers are trained separately from the other layers which allows the other layers to be trained in a greedily layer by layer from bottom to up. All other layers are trained after pre-training [136]. The sub-classes of generative models are Recurrent Neural Network (RNN), Deep Auto-Encoder, Deep Boltzmann Machine (DBM) and Deep Believe networks (DBN).

- Recurrent Neural Network (RNN)

  RNN is a class of deep networks that are either considered supervised or unsupervised learning with an input sequential data whose length could be as large as its depth [139]. The RNN model architecture is a feedback loop linking layer by layer with the ability to store data of previous input increasing the reliability of the model [140]. There are two types on the RNN in terms of architecture: Elman and Jordan RNNs. The Elman model has a simple feedback looping layer by layer. The Jordan model has

a feedback looping all neurons within a layer to the next layer. There exists also a feedback connecting a neuron to itself. The ability of the Jordan RNN to store information in the neurons allows it to train less input vector for classification of normal and abnormal patterns with high accuracy [140]. Figures 4.4 and 4.5 are simple architectures of the Elman RNN and Jordan RNN showing the context unit known to store information of the previous output of hidden layer.
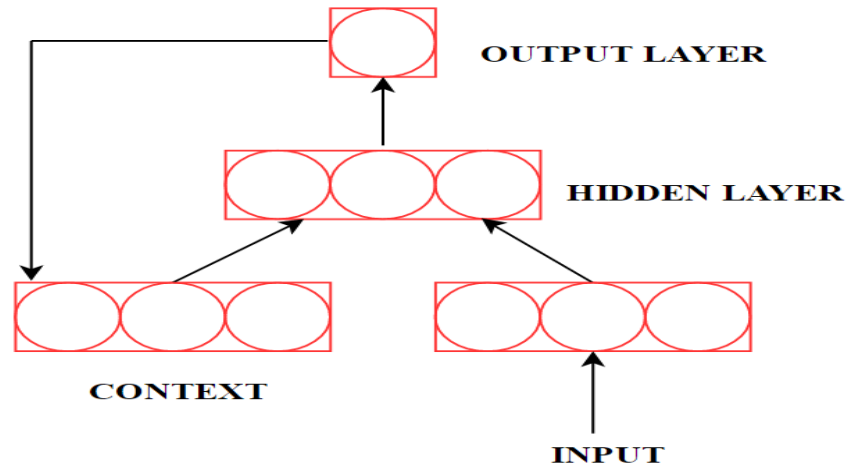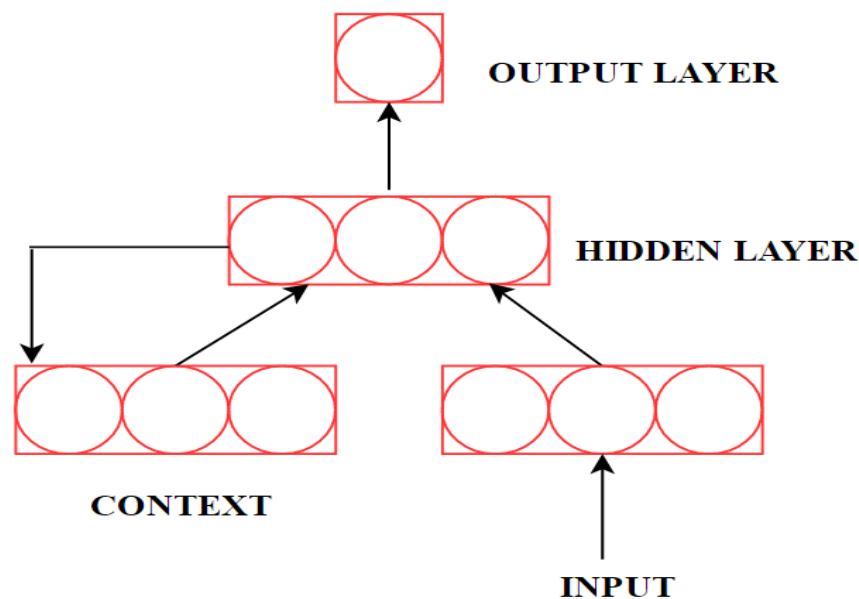

Figure 4.4 Jordan Network

Figure 4.5 Elman Network

A study by K. Jihyun *et al*. [141] applied long short term memory (LSTM) architecture to RNN and trained the IDS using KDD Cup '99 dataset. By comparing the accuracy with other IDS classifiers, LSTM-RNN recorded 96.93% accuracy with a detection rate of 98.88%. Although the FAR was slightly higher than the others, they concluded its overall performance was the best.

- Deep Auto-Encoder
  These are energy based deep models classified as generative models in their original form. It comes in different forms mostly also generative models. Other forms are stacked auto encoder and de noising auto encoder a [138].

An Auto-Encoder becomes deep when it has multiple hidden layers. It is made up of an input layer unit representing the sample data, one or two hidden layer units where the features are transformed and then mapped to the output layer unit for reconstruction. Training the auto-encoder gives it a "bottleneck" structure where the hidden layer becomes narrower than the input layer to prevent the model from learning its identity function [120].
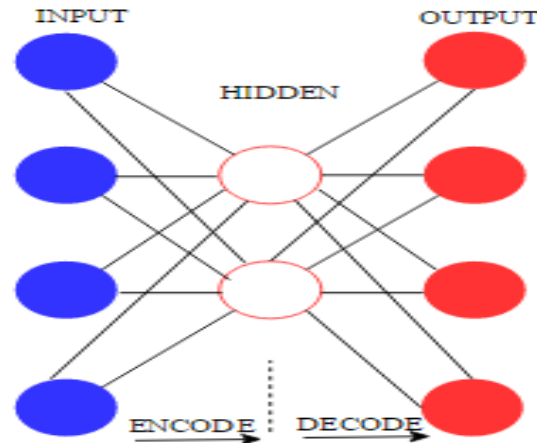

Figure 4.6 Deep auto encoder

Unfortunately, the deep auto encoder when trained with back propagation has not been a success the evaluation gets stuck in local minima with a minimal gradient signal when trained with back propagation. Pre-training a deep auto encoder using the greedy layer wise approach by training each of the layers in turns has proven to alleviate the backpropagation problems [120], [139],[142].

In [143], B. Abolhasanzadeh proposed an approach to detect attacks in big data using deep auto encoder. The experiment was conducted on NSL-KDD data set to test the method of applying bottle neck features in dimensionality reduction as part of intrusion detection. The results in terms of accuracy rate out performed PCA, factor analysis and Kernel/PCA. It was concluded; the results recorded in terms of accuracy makes this approach promising one for real world intrusion detection.

- Deep Boltzmann Machine(DBM)

  DBM is a unidirectional graphical model. Currently there exist no connection between units on the same layer but between the input units and the hidden units. DBM when trained with a large supply of unlabelled data and fine-tuned with labelled data acts as a good classifier [136]. Its structure is an offspring of a general Boltzmann machine (BM) which is a network of units based on stochastic decisions to determine their on and off states [138]. BM algorithm is simple to train but turns to be slow in the process. A reduction in the number of hidden layers of a DBM to one forms a Restricted Boltzmann Machine (RBM) [139]. DBM when trained with a large supply of unlabelled data and fine-tuned with labelled data acts a good classifier. Training a stack of RBM with many hidden layers using the feature activation on one RBM as the input for the next layer leads to the formation of Deep Believe Network (DBN).
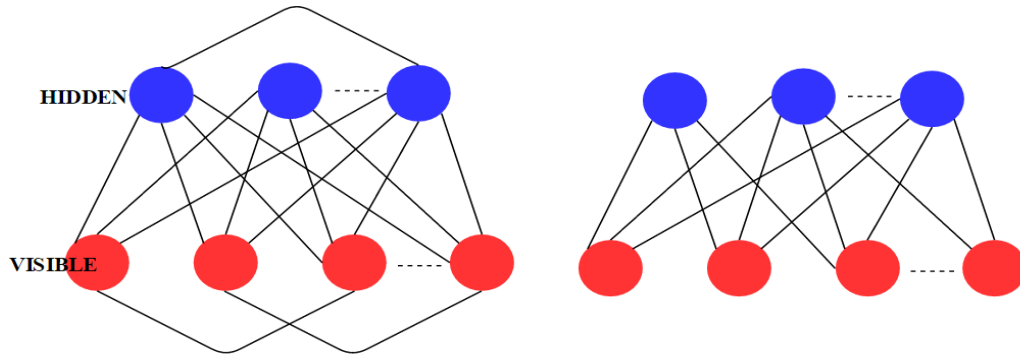
Figure 4.7 Left: A Boltzmann machine with stochastic on and off hidden features and a variable layer representing a vector of stochastic on and off states. Right: An RBM with no visible to visible units connected and no hidden to hidden units connected.

U. Fiore *et al*. in [144]explored RBM in anomaly detection by training a network with real world data traces from a 24hour work station traffic. This experiment was to test the accuracy of RBM to classify normal data and data infected by bot. a second experiment trained RBM with KDD Cup '99 data set and tested against real world data. To randomize the order of test data, the experiment was repeated 10times. The experiment confirmed testing a classifier in two different networks training data affects the performance. They suggested the nature of anomalous traffic and normal traffic should be investigated.

- Deep Belief Networks (DBN)

  DBN uses both unsupervised pre training and supervised fine-tuning techniques to construct the models. Figure 4.8 shows a DBN which is made up of a stack of Restricted Boltzmann Machines (RBMs) and one or more additional layers for discrimination task. RBMs are probabilistic generative models that learn a joint probability distribution of observed (training) data without using data labels. Once the structure of a DBN is determined the goal for training is to learn the weights *w* between layers. Each node is independent of other nodes in the same layer given all nodes which gives it the characteristic allowing us to train the generative weights of each RBM [120]. It then goes through a greedy layer by layer learning algorithm which learns each stack of RBM's layer at a time. In Figure 4.8 left, the top layers in red form a RBM and the lower layers in blue form directed sigmoid believe network [145].
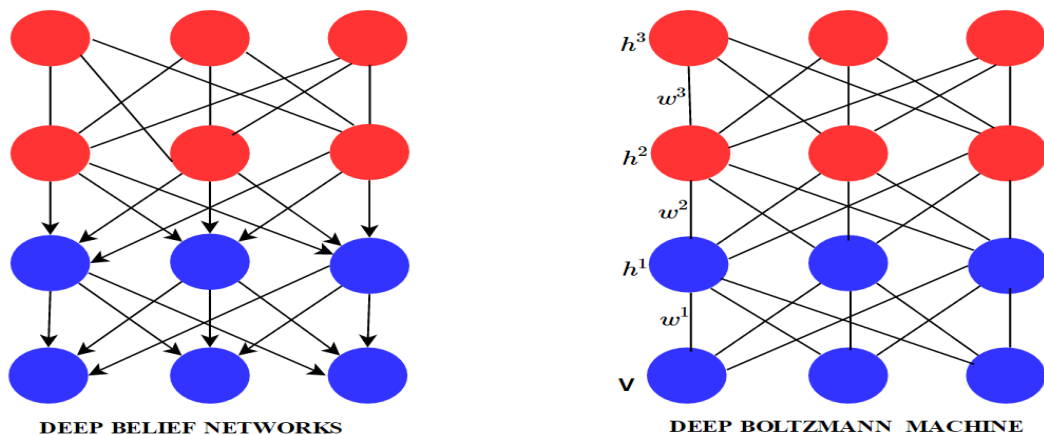

Figure 4.8 Three layer Deep Belief network and three layers deep Boltzmann machine

DBN has been trained as a classifier by N. Gao *et al*. [146] to detect intrusion by comparing the performance to SVM and ANN. The classifiers were trained on KDD data set. The authors proved that deep learning of DBN can successfully be used as an effective ID. They concluded the greedy layer by layer learning algorithm when used to pre-train and fine-tune a DBN gives a high accuracy in classification. The results showed that DBN recorded the best accuracy of 93.49%, a TP value of 92.33 and FP of 0.76%.

Z. Alom *et al*.[147] also exploited the DBNs capabilities to detect intrusion through series of experiments. The authors trained DBN with NSL-KDD data to identify unknown attack on it. They concluded by proposing DBN as a good IDS based on an accuracy of 97.5% achieved in the experiment. This results was compared with existing DBN-SVM and SVM classifiers which it out performed.

4.2.1 Discriminative Architecture

The discriminative architecture uses discriminative power for classification by characterising the posterior distributions of classes conditioned on the input data. Recurrent neural network and convolutional neural network are two types of discriminative architecture.

- Recurrent neural network (RNN) uses discriminative power for classification when the model's output is an explicit labelled data in sequence with the input data sequence. To train RNN as a discriminative model, training data needs to be pre-segmented and a post-processing to transform the output to a labelled data [138].

- Convolutional neural network

  A convolutional neural network (CNN) is a type of discrimination deep architecture with one or more convolutional and pooling layers in an array to form a multilayer neural network [139],[148],[149]. In general, convolutional layers share many weights followed by sampling of the convolutional layer's output by the pooling layer which results in some form of translational invariant properties [139].

  CNN has fewer parameters as compared to other connected networks with the same number of hidden units which gives it an advantage of easier training [148]. CNN architecture is that of multi-layer perceptron [150], and are variant of MLP which are inspired biologically. Hubel and Wiesel worked on the cat's visual cortex and deduced that visual cortex is made up of an arrangement of cells in a complex manner. These cells are sensitive to small sub-regions of the visual field known as the receptive field. These fields are positioned to shield the entire visual field to enable the cell behave just like a local filter over the input space [151]. The series of layers making up a CNN architecture are the convolutional layer, max pooling layer and the fully connected layer [150], [152]. The convolutional layer is made up of neurons forming a rect6angular grid where previous layers are made of neuros shaped as a rectangular grid. The rectangular grid neurons are connected to each other with the inputs from previous rectangular units through a set of weights known as filter banks [149], [152]. These weights for the rectangular units do not change for every rectangular grid of neuron to form convolutional layers. In architectures where the convolutional layer is made up of different grids [152], each grid uses a different filter bank [149], [153]. Each convolutional layer is followed by a pooling layer which

merges subsets of the convolutional layer's rectangular block by taking sub-samples to give an output of the block. The pooling can be done in several ways such as [152] computing the maximum or average or a learned linear summing of neurons in the blocks. Some blocks turn to be shifted more than a row or a column and in turn feed in an s input to neighbouring pooling units. This causes a reduction in the dimension of the system design and thus causing variations to the input [149], [153]. The final stage which has several convolutional and max-pooling layers non-linearly stacked in the neural network form a fully connected layer of network [152]. The connectivity which allows the set of weights of the filter banks to be trained easily [149], [152].
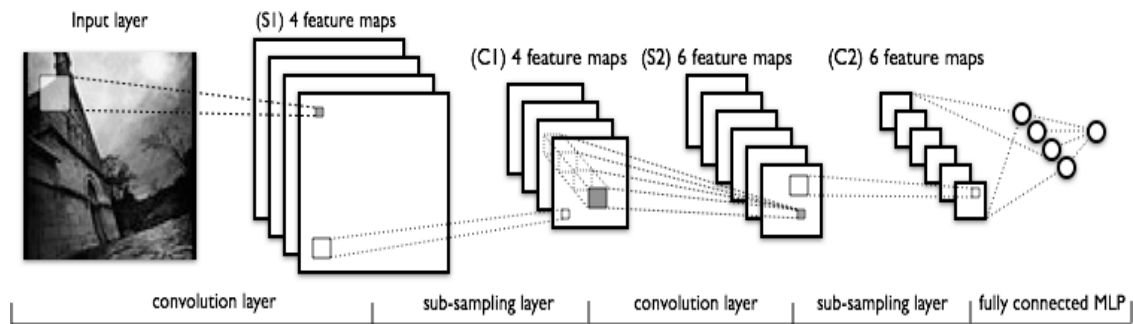

Figure 4.9 Image of CNN showing layers making up architecture [154]

5.0 Conclusion

Shallow and deep networks intrusion detection systems have gained a considerable interest commercially and amongst the research community. With advancement in data sizes, intrusion detection systems should have the characteristics to handle noisy data with high accuracy in detection with high computational speed. This paper gives an overview of the general classification of intrusion detection systems and taxonomy with recent and past works. This taxonomy gives a clear description of intrusion detection system and its complexity.

Current studies of deep learning intrusion detection systems have been reviewed in this paper to help address the challenges in this new technique still in its early stages in intrusion detection. In particular recent papers have been reviewed in this work considering all the machine learning techniques including the single and hybrid techniques.

The scope of the work on classifying intrusion detection systems, reviewing the various methods of detecting anomaly, performance of these methods were based on past and recent works revealing the advantages and disadvantages of each of them.

The focus of the paper on shallow and deep networks described experiments comparing the performance of these learning algorithms. The experiments demonstrated deep networks significantly outperformed the shallow network in detection of attacks.

To the best of our knowledge CNN has not been exploited in the field of intrusion detection but has proven to be a good classifier. DBN is also new in its exploitation in this field and experimental works are still in progress to determine the reliability of these learning algorithms to detect attacks.

Signature based technique have been in use commercially but have not been able to detect all types of attacks especially if the IDS signature list did not contain the right signature.

Research work is in progress experimenting new approaches to test the reliability and efficiency of knowledge based and behavioural approaches in intrusion detection.

REFERENCE

[1]   N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, and M. C. Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection," in *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, 2016, pp. 1–6.

[2]   B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[3]   SANS Institute, "The History and Evolution of Intrusion Detection." [Online]. Available: https://www.sans.org/reading-room/whitepapers/detection/history-evolution-intrusion-detection-344. [Accessed: 20-Feb-2016].

[4]   R. P. R. I. Sravan Kumar Jonnalagadda, "A Literature Survey and Comprehensive Study of Intrusion Detection," *Int. J. Comput. Appl.*, vol. 11, no. 81(16), pp. 40–47, 2013.

[5]   R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–29, Mar. 2014.

[6]   "Internet of Things: How Much are We Exposed to Cyber Threats? - InfoSec Resources." [Online]. Available: http://resources.infosecinstitute.com/internet-things-much-exposed-cyber-threats/. [Accessed: 10-Dec-2015].

[7]   B. W. Masduki, K. Ramli, F. A. Saputra, and D. Sugiarto, "Study on implementation of machine learning methods combination for improving attacks detection accuracy on Intrusion Detection System (IDS)," in *2015 International Conference on Quality in Research (QiR)*, 2015, pp. 56–64.

[8]   P. Sneath and R. R. Sokal, "Numerical taxonomy. The principles and practices of numerical classification," 1973. [Online]. Available: http://www.brclasssoc.org.uk/books/Sneath/title_contents.pdf.

[9]   M. A., "SIMPSON, GG Principles of animal taxonomy. Columbia University Press, New York: 1990. Pp xii, 247; illustrated. Price: US $21.50. ISBN: 0-231-09650-X (paperback reprint)," *Arch. Nat. Hist.*, vol. 19, no. 1, pp. 124--124, 1992.

[10]   S. Axelsson and Stefan, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proceedings of the 6th ACM conference on Computer and communications security - CCS '99*, 1999, pp. 1–7.

[11]   J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[12]   D. Rozenblum, "Understanding Intrusion Detection Systems," *SANS Inst.*, no. 122, pp. 11–15, 2001.

[13]   "What it is Network intrusion detection system? | COMBOFIX." [Online]. Available: http://www.combofix.org/what-it-is-network-intrusion-detection-system.php. [Accessed: 10-Dec-2015].

[14] H. Debar, M. Dacier, and A. Wespi, "A revised taxonomy for intrusion-detection systems," *Ann. Des Télécommunications*, vol. 55, no. 7–8, pp. 361–378.

[15] S. Axelsson and S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.6603. [Accessed: 30-Sep-2016].

[16] A. Hafez, S. and hamilton Jr, and A. John, "Intrusion Detection Systems (IDS) Taxonomy-A Short Review," *This is a Paid Advert. STN 13-2 June 2010 Defensive Cyber Secur. Policies Proced. 2*, p. 23.

[17] C. Xenakis, C. Panos, and I. Stavrakakis, "A comparative evaluation of intrusion detection architectures for mobile ad hoc networks," *Comput. Secur.*, vol. 30, no. 1, pp. 63–80, 2011.

[18] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[19] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Comput. Networks*, vol. 31, no. 8, pp. 805–822, Apr. 1999.

[20] Sans Penetration Testing, "Host- vs. Network-Based Intrusion Detection Systems," 2001. [Online]. Available: https://cyber-defense.sans.org/resources/papers/gsec/host-vs-network-based-intrusion-detection-systems-102574. [Accessed: 24-Feb-2016].

[21] SANS Institute InfoSec Reading Room, "Application of Neural Networks to Intrusion Detection," 2001. [Online]. Available: https://www.sans.org/reading-room/whitepapers/detection/application-neural-networks-intrusion-detection-336. [Accessed: 24-Feb-2016].

[22] A. Mounji and B. Le Charlier, "Continuous assessment of a Unix configuration: integrating intrusion detection and configuration analysis," in *Proceedings of SNDSS '97: Internet Society 1997 Symposium on Network and Distributed System Security*, 1997, pp. 27–35.

[23] S. S. Soniya and S. M. C. Vigila, "Intrusion detection system: Classification and techniques," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016, pp. 1–7.

[24] R. P. R. I. Sravan Kumar Jonnalagadda, "A Literature Survey and Comprehensive Study of Intrusion Detection," *Int. J. Comput. Appl.*, vol. 81, no. 16, pp. 40--47, 2013.

[25] Heba Fathy Ahmed Mohamed Eid, "Computational Intelligence in Intrusion Detection System," 2013. [Online]. Available: http://scholar.cu.edu.eg/sites/default/files/abo/files/phd_thesis_computational_intelligence_in_intrusion_detection_system_2013.pdf. [Accessed: 24-Feb-2016].

[26] S. Mallissery, J. Prabhu, and R. Ganiga, "Survey on intrusion detection methods," in *3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011)*, 2011, pp. 224–228.

[27] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[28] S. S. Tirumala, H. Sathu, and A. Sarrafzadeh, "Free and open source intrusion detection systems: A study," in *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2015, vol. 1, pp. 205–210.

[29] S. Liu, J. Gong, J. Chen, Y. Peng, W. Yang, W. Zhang, and A. Jakalan, "A flow based method to detect penetration," in *The 7th IEEE/International Conference on Advanced Infocomm Technology*, 2014, pp. 184–191.

[30] H. Kozushko, "Intrusion detection: Host-based and network-based intrusion detection systems," *Sept.*, vol. 11, 2003.

[31] N. K. Mittal, "A survey on Wireless Sensor Network for Community Intrusion Detection Systems," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 107–111.

[32] J. Shun and H. a. Malki, "Network Intrusion Detection System Using Neural Networks," *2008 Fourth Int. Conf. Nat. Comput.*, vol. 5, pp. 242–246, 2008.

[33] A. G. Tokhtabayev and V. A. Skormin, "Non-Stationary Markov Models and Anomaly Propagation Analysis in IDS," in *Third International Symposium on Information Assurance and Security*, 2007, pp. 203–208.

[34] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.

[35] A. Qayyum, M. H. Islam, and M. Jamil, "Taxonomy of statistical based anomaly detection techniques for intrusion detection," in *Proceedings of the IEEE Symposium on Emerging Technologies, 2005.*, 2005, pp. 270–276.

[36] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, Feb. 2009.

[37] T. R. Pillai, S. Palaniappan, A. Abdullah, and H. M. Imran, "Predictive modeling for intrusions in communication systems using GARMA and ARMA models," in *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, 2015, pp. 1–6.

[38] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 1992, pp. 240–250.

[39] A. M. Richard Heady,George Luger, "The Architecture of a Network Level Intrusion Detection System," *Department of Computer Science ,College of Engineering,University of New Mexico*, 1990. [Online]. Available: https://www.researchgate.net/profile/Mark_Servilla/publication/242637613_The_archi tecture_of_a_network_level_intrusion_detection_system/links/5564805a08ae06101ab df482.pdf. [Accessed: 19-Feb-2016].

[40] K. Wang and S. J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," Springer Berlin Heidelberg, 2004, pp. 203–222.

[41] T. F. Lunt, "A survey of intrusion detection techniques," *Comput. Secur.*, vol. 12, no. 4, pp. 405–418, Jun. 1993.

[42] R. China Appala Naidu and P. S. Avadhani, "A comparison of data mining techniques for intrusion detection," in *2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2012, pp. 41–44.

[43] D. Anderson, T. Frivold, A. Tamaru, A. Valdes, and B. Release, "Next Generation Intrusion Detection Expert System (NIDES), Software Users Manual." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.5048. [Accessed: 19-Feb-2016].

[44] D. E. Denning, "An Intrusion-Detection Model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[45] K. M. P. V. Jyothsna, V. V. Rama Prasad, "A Review of Anomaly based Intrusion Detection Systems," *Int. J. Comput. Appl.*, vol. 28, no. 7, pp. 26–35, 2011.

[46] A. B. Ashfaq, M. Javed, S. A. Khayam, and H. Radha, "An Information-Theoretic Combining Method for Multi-Classifier Anomaly Detection Systems," in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

[47] B. A. Kuperman, "A categorization of computer security monitoring systems and the impact on the design of audit sources," *Purdue Univ. West Lafayette,* 2004.

[48] W. Sha, Y. Zhu, T. Huang, M. Qiu, Y. Zhu, and Q. Zhang, "A Multi-order Markov Chain Based Scheme for Anomaly Detection," in *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, 2013, pp. 83–88.

[49] Y. Nong, "A markov chain model of temporal behavior for anomaly detection," in *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop,vol. 166*, 2000, p. 169.

[50] H. Zhengbing, L. Zhitang, and W. Junqi, "A Novel Network Intrusion Detection System (NIDS) Based on Signatures Search of Data Mining," in *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)*, 2008, pp. 10–16.

[51] H. E. Poston, "A brief taxonomy of intrusion detection strategies," in *2012 IEEE National Aerospace and Electronics Conference (NAECON)*, 2012, pp. 255–263.

[52] P. A. Porras and R. A. Kemmerer, "Penetration state transition analysis: A rule-based intrusion detection approach," in *[1992] Proceedings Eighth Annual Computer Security Application Conference*, 1992, pp. 220–229.

[53] T. Verwoerd and R. Hunt, "Intrusion detection techniques and approaches," *Comput. Commun.*, vol. 25, no. 15, pp. 1356–1365, 2002.

[54] ZhiSong Pan, Hong Lian, GuYu Hu, and GuiQiang Ni, "An integrated model of intrusion detection based on neural network and expert system," in *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, 2005, p. 2 pp.-pp.672.

[55] SANS Institute InfoSec Reading Room, "Intrusion Detection Systems: An Overview of RealSecure," 2001. [Online]. Available: http://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-overview-realsecure-342. [Accessed: 04-Mar-2016].

[56] "IBM X-Force: Ahead of the Threat - Resources," 07-Dec-2011. [Online]. Available: http://www-03.ibm.com/security/xforce/resources.html#all. [Accessed: 04-Mar-2016].

[57] Sandeep Kumar, "CLASSIFICATION AND DETECTION OF COMPUTER INTRUSIONS," 1995. [Online]. Available: http://web.mst.edu/~tauritzd/compsec/papers/kumar95classification.pdf. [Accessed: 04-Mar-2016].

[58] T.-F. Sheu, N.-F. Huang, and H.-P. Lee, "NIS04-6: A Time- and Memory- Efficient String Matching Algorithm for Intrusion Detection Systems," in *IEEE Globecom 2006*, 2006, pp. 1–5.

[59] C. Vaid and H. K. Verma, "Anomaly-based IDS implementation in cloud environment using BOAT algorithm," in *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, 2014, pp. 1–6.

[60] K. M. A. Alheeti, A. Gruebler, K. D. McDonald-Maier, and A. Fernando, "Prediction of DoS attacks in external communication for self-driving vehicles using a fuzzy petri net model," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, 2016, pp. 502–503.

[61] J. Hong, C.-C. Liu, and M. Govindarasu, "Detection of cyber intrusions using network-based multicast messages for substation automation," in *ISGT 2014*, 2014, pp. 1–5.

[62] R. Mohan, V. Vaidehi, and S. S. Chakkaravarthy, "Complex Event Processing based Hybrid Intrusion Detection System," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, pp. 1–6.

[63] S. H. Vasudeo, P. Patil, and R. V. Kumar, "IMMIX-intrusion detection and prevention system," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, 2015, pp. 96–101.

[64] W. Haider, J. Hu, X. Yu, and Y. Xie, "Integer Data Zero-Watermark Assisted System Calls Abstraction and Normalization for Host Based Anomaly Detection Systems," in *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, 2015, pp. 349–355.

[65] H. Toumi, M. Talea, K. Sabiri, and A. Eddaoui, "Toward a trusted framework for cloud computing," in *2015 International Conference on Cloud Technologies and Applications (CloudTech)*, 2015, pp. 1–6.

[66] M. Guerroumi, A. Derhab, and K. Saleem, "Intrusion Detection System against Sink Hole Attack in Wireless Sensor Networks with Mobile Sink," in *2015 12th International Conference on Information Technology - New Generations*, 2015, pp. 307–313.

[67] D. Narsingyani and O. Kale, "Optimizing false positive in anomaly based intrusion detection using Genetic algorithm," in *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 2015, pp. 72–77.

[68] W. Haider, J. Hu, and M. Xie, "Towards reliable data feature retrieval and decision engine in host-based anomaly detection systems," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 513–517.

[69] N. B. Aissa and M. Guerroumi, "A genetic clustering technique for Anomaly-based Intrusion Detection Systems," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed*

Computing (SNPD), 2015, pp. 1–6.

[70]   O. Hamdi, M. Mbaye, and F. Krief, "A cloud-based architecture for network attack signature learning," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, 2015, pp. 1–5.

[71]   Xiaodong Lin and Rongxing Lu, "Vehicular Ad Hoc Network Security and Privacy," 21-Aug-2015. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7159960. [Accessed: 03-Jun-2016].

[72]   S. Banerjee, R. Nandi, R. Dey, and H. N. Saha, "A review on different Intrusion Detection Systems for MANET and its vulnerabilities," in *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, 2015, pp. 1–7.

[73]   P. Satam, "Cross Layer Anomaly Based Intrusion Detection System," in *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, 2015, pp. 157–161.

[74]   K. Patel and Kayur, "Lowering the barrier to applying machine learning," in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10*, 2010, p. 2907.

[75]   E. Alpaydın, "Introduction To Machine learning," 2010. [Online]. Available: https://www.lri.fr/~xlzhang/KAUST/CS229_slides/chapter18_RL.pdf. [Accessed: 20-Jan-2015].

[76]   A. A. Ghorbani, W. Lu, and M. Tavallaee, "Network Intrusion Detection and Prevention," *... and Autonomous System*, 2010. [Online]. Available: http://link.springer.com/10.1007/978-0-387-88771-5.

[77]   H. T. Nguyen, K. Franke, and S. Petrovic, "Feature Extraction Methods for Intrusion Detection Systems," in *Threats, Countermeasures, and Advances in Applied Information Security*, vol. 3, IGI Global, 2012, pp. 23–52.

[78]   A. K. Jain, P. W. Duin, and J. Jianchang Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[79]   C. Kruegel, D. Mutz, F. Valeur, and G. Vigna, "On the Detection of Anomalous System Call Arguments," Springer Berlin Heidelberg, 2003, pp. 326–343.

[80]   M. V. Mahoney and P. K. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection," vol. 2820, 2003, pp. 220–237.

[81]   W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 227–261, Nov. 2000.

[82]   M. Khosravi-Farmad, A. A. Ramaki, and A. G. Bafghi, "Risk-based intrusion response management in IDS using Bayesian decision networks," in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2015, pp. 307–312.

[83]   R. Stuart, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River,New Jersey: Prentice Hall Pa, 2009.

[84]   A. R. Onik, N. F. Haq, and W. Mustahin, "Cross-breed type Bayesian network based intrusion detection system (CBNIDS)," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*, 2015, pp. 407–412.

[85]   M. A. Bode, S. A. Oluwadare, B. K. Alese, and A. F.-B. Thompson, "Risk analysis in cyber situation awareness using Bayesian approach," in *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2015, pp. 1–12.

[86]   Hartmut Pohlheim, "Competition and Cooperation in Extended Evolutionary Algorithms," 2001. [Online]. Available: http://www.pohlheim.com/Papers/conf_gecco2001/PohlheimH_CompetitionExtEA_G ECCO2001.pdf. [Accessed: 10-Mar-2016].

[87]   W. Li, "Using Genetic Algorithm for Network Intrusion Detection," *Proc. United States Dep. Energy Cyber Secur. Grou*, pp. 1--8, 2004.

[88]   Tao Xia, Guangzhi Qu, S. Hariri, and M. Yousif, "An efficient network intrusion detection method based on information theory and genetic algorithm," in *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pp. 11–17.

[89]   M. Padmadas, N. Krishnan, J. Kanchana, and M. Karthikeyan, "Layered approach for intrusion detection systems based genetic algorithm," in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, 2013, pp. 1–4.

[90]   G. Wang, D.-Y. Yeung, and F. H. Lochovsky, "A kernel path algorithm for support vector machines," in *Proceedings of the 24th international conference on Machine learning - ICML '07*, 2007, pp. 951–958.

[91]   C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167.

[92]   F. Zhu, N. Ye, D. Pan, and W. Ding, "Incremental Support Vector Machine Learning: An Angle Approach," in *2011 Fourth International Joint Conference on Computational Sciences and Optimization*, 2011, pp. 288–292.

[93]   B. Senthilnayaki, K. Venkatalakshmi, and A. Kannan, "Intrusion detection using optimal genetic feature selection and SVM based classifier," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, pp. 1–4.

[94]   L. Teng, S. Teng, F. Tang, H. Zhu, W. Zhang, D. Liu, and L. Liang, "A Collaborative and Adaptive Intrusion Detection Based on SVMs and Decision Trees," in *2014 IEEE International Conference on Data Mining Workshop*, 2014, pp. 898–905.

[95]   K. Shi, L. Li, H. Liu, J. He, N. Zhang, and W. Song, "An improved KNN text classification algorithm based on density," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 2011, pp. 113–117.

[96]   Y. Canbay and S. Sagiroglu, "A Hybrid Method for Intrusion Detection," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 156–161.

[97]   H. Zhang and G. Chen, "The Research of Face Recognition Based on PCA and K-Nearest Neighbor," in *2012 Symposium on Photonics and Optoelectronics*, 2012, pp.

1–4.

[98]   Q. Zeng and S. Wu, "Anomaly Detection Based on Multi-Attribute Decision," in *2009 WRI Global Congress on Intelligent Systems*, 2009, pp. 394–398.

[99]   K. A. Jalill, M. H. Kamarudin, and U. T. Mara, "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion," pp. 221–226, 2010.

[100]   F. Sebastiani and Fabrizio, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[101]   T. Komviriyavut, P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Network intrusion detection and classification with Decision Tree and rule based approaches," in *2009 9th International Symposium on Communications and Information Technology*, 2009, pp. 1046–1050.

[102]   S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 Decision Tree," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 2023–2026.

[103]   S. Rajasekaran and G. A. V. Pai, "Neural Neworks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications (WITH CD)," 2003. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=bVbj9nhvHd4C&pgis=1. [Accessed: 29-Mar-2016].

[104]   M. Wahengbam and N. Marchang, "Intrusion Detection in MANET using fuzzy logic," in *2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, 2012, pp. 189–192.

[105]   A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," 2007.

[106]   H. Om and A. Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system," in *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, 2012, pp. 131–136.

[107]   S. Gupta, "An effective model for anomaly IDS to improve the efficiency," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 190–194.

[108]   A. Ben Ayed, M. Ben Halima, and A. M. Alimi, "Survey on clustering methods: Towards fuzzy clustering for big data," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 2014, pp. 331–336.

[109]   Li Jun Tao, Liu Yin Hong, and Hao Yan, "The improvement and application of a K-means clustering algorithm," in *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2016, pp. 93–96.

[110]   P. Jongsuebsuk, N. Wattanapongsakorn, and C. Charnsripinyo, "Network intrusion detection with Fuzzy Genetic Algorithm for unknown attacks," in *The International Conference on Information Networking 2013 (ICOIN)*, 2013, pp. 1–5.

[111]   A.-C. Enache and V. Sgarciu, "Anomaly Intrusions Detection Based on Support Vector Machines with an Improved Bat Algorithm," in *2015 20th International Conference on Control Systems and Computer Science*, 2015, pp. 317–321.

[112] S. Akbar, J. A. Chandulal, K. N. Rao, and G. S. Kumar, "Improving network security using machine learning techniques," in *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 2012, pp. 1–5.

[113] A. S. A. Aziz, A. E. Hassanien, S. E.-O. Hanaf, and M. F. Tolba, "Multi-layer hybrid machine learning techniques for anomalies detection and classification approach," in *13th International Conference on Hybrid Intelligent Systems (HIS 2013)*, 2013, pp. 215–220.

[114] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015.

[115] A. A. Aburomman and M. Bin Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, Jan. 2016.

[116] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, Jan. 2011.

[117] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System," in *2016 3rd International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[118] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, Jan. 2010.

[119] S. Dong, D. Zhou, and W. Ding, "The Study of Network Traffic Identification Based on Machine Learning Algorithm," *2012 Fourth Int. Conf. Comput. Intell. Commun. Networks*, pp. 205–208, Nov. 2012.

[120] K. Murphy, "Machine learning: a probabilistic perspective," *Chance encounters: Probability in …*, 2012. [Online]. Available: http://link.springer.com/chapter/10.1007/978-94-011-3532-0_2. [Accessed: 06-Jan-2015].

[121] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[122] P. Barapatre, N. Z. Tarapore, S. G. Pukale, and M. L. Dhore, "Training MLP neural network to reduce false alerts in IDS," in *2008 International Conference on Computing, Communication and Networking*, 2008, pp. 1–7.

[123] J. Bi, K. Zhang, and X. Cheng, "Intrusion Detection Based on RBF Neural Network," in *2009 International Symposium on Information Engineering and Electronic Commerce*, 2009, pp. 357–360.

[124] C. Zhang, J. Jiang, and M. Kamel, "Comparison of BPL and RBF Network in Intrusion Detection System," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 466–470.

[125] M. Kamel, "RBF-based real-time hierarchical intrusion detection systems," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, 2003, vol. 2, pp. 1512–1516.

[126] Y. Fu, Y. Zhu, and H. Yu, "Study of Neural Network Technologies in Intrusion Detection Systems," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009, pp. 1–4.

[127] A. N. Z.-H. Peter Lichodzijewski, "Host-Based Intrusion Detection Using Self-organizing Maps," in *IJCNN '02*, 2002, pp. 1714–1719.

[128] M. I. H. H. Gunes Kayacik, A. Nur Zincir-Heywood, "A hierarchical SOM-based intrusion detection system," vol. 20, pp. 439–451, 2007.

[129] V. Dinesh Kumar and S. Radhakrishnan, "Intrusion detection in MANET using Self Organizing Map (SOM)," in *2014 International Conference on Recent Trends in Information Technology*, 2014, pp. 1–8.

[130] M. Chauhan, A. Pratap, and A. Dixit, "Designing a technique for detecting intrusion based on modified Adaptive Resonance Theory Network," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 448–451.

[131] S. Haykin, "Neural Networks-A comprehensive foundation," 1999. [Online]. Available: https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Neural Networks - A Comprehensive Foundation - Simon Haykin.pdf. [Accessed: 22-Jan-2015].

[132] P. Somwang and W. Lilakiatsakun, "Intrusion detection technique by using fuzzy ART on computer network security," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2012, pp. 697–702.

[133] A. S. A. Iftikhar Ahmad, Azween B Abdullah, "Evaluating Neural Network Intrusion Detection Approaches Using Analytic Hierarchy Process," in *2010 IEEE International Symposium on Information Technology*, 2010.

[134] and Y. T. G. E. Hinton, S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[135] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," Santa Cruz, 1994.

[136] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends® in Machine Learning*, 2009. [Online]. Available: http://www.nowpublishers.com/product.aspx?product=MAL&doi=2200000006. [Accessed: 10-Jul-2014].

[137] M. I. Jordan, "Learning in Graphical Models.," 1998. [Online]. Available: https://www.cs.cmu.edu/~tom/10-702/zoubin-varintro.pdf. [Accessed: 02-Aug-2016].

[138] Li Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Trans. Signal Inf. Process.*, vol. 3, no. e2, pp. 1--29, 2014.

[139] L. Deng and D. Yu, "Foundations and Trends® in Signal Processing," 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/. [Accessed: 02-Aug-2016].

[140] L. O. Anyanwu, J. Keengwe, and G. A. Arome, "Scalable Intrusion Detection with Recurrent Neural Networks," in *IEEE 2010 Seventh International Conference on*

Information Technology: New Generations, 2010, pp. 919–923.

[141] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," in *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5.

[142] A. Ng, "Stacked Autoencoders - Ufldl." [Online]. Available: http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders. [Accessed: 18-Feb-2015].

[143] B. Abolhasanzadeh, "Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, 2015, pp. 1–5.

[144] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, Dec. 2013.

[145] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," *Artif. Intell.*, vol. 5, no. 2, pp. 448–455, 2009.

[146] N. Gao, L. Gao, Q. Gao, and H. Wang, "An Intrusion Detection Model Based on Deep Belief Networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, 2014, pp. 247–252.

[147] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *2015 National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 339–344.

[148] "Unsupervised Feature Learning and Deep Learning Tutorial." [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/. [Accessed: 13-Nov-2015].

[149] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[150] M. Dalto, "Deep neural networks for time series prediction with applications in ultra-short-term wind forecasting." [Online]. Available: http://www.fer.unizg.hr/_download/repository/KDI-Djalto.pdf. [Accessed: 19-Feb-2015].

[151] L. Lab, "Deep Learning Tutorial," 2015. [Online]. Available: http://deeplearning.net/tutorial/deeplearning.pdf. [Accessed: 20-Aug-2016].

[152] "Convolutional Neural Networks - Andrew Gibiansky." [Online]. Available: http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/. [Accessed: 24-Nov-2015].

[153] Y. LeCun, "Learning Invariant feature Hierarchies," 2012. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-eccv-12.pdf. [Accessed: 13-Nov-2015].

[154] "Convolutional Neural Networks (LeNet) — DeepLearning 0.1 documentation." [Online]. Available: http://deeplearning.net/tutorial/lenet.html. [Accessed: 29-Apr-2016].